ESaaSA 2015

2nd INTERNATIONAL WORKSHOP ON EMERGING SOFTWARE AS A SERVICE AND ANALYTICS

Victor Chang, Muthu Ramachandran,
Gary Wills, Robert Walters, Verena Kantere
and Chung-Sheng Li (Eds.)

Proceedings of ESaaSA 2015
2nd International Workshop on Emerging Software as a Service and Analytics

In conjuction with the 5th International Conference on Cloud Computing and Services Science - CLOSER 2015

Lisbon, Portugal | **May, 2015**

ESaaSA 2015

Proceedings of the
2nd International Workshop on Emerging Software as a Service and Analytics

Lisbon, Portugal

20 - 22 May, 2015

Edited by Victor Chang, Muthu Ramachandran, Gary Wills,
Robert Walters, Verena Kantere and Chung-Sheng Li

Printed in Portugal

ISBN: 978-989-758-110-6

Depósito Legal: 391664/15

http://closer.scitevents.org/ESaaSA.aspx

closer.secretariat@scitevents.org

Brief Contents

WORKSHOP CHAIRS

Victor Chang, Leeds Beckett University, U.K.

Muthu Ramachandran, Leeds Beckett University, U.K.

Gary Wills, University of Southampton, U.K.

Robert Walters, University of Southampton, U.K.

Verena Kantere, University of Geneva, Switzerland

Chung-Sheng Li, IBM, U.S.A.

PROGRAM COMMITTEE

Omar Abdul-Rahman, National Institute of Informatics, Japan

Saad Alahmari, Princess Noura Bint Abdulrahman University, Saudi Arabia

Naif Aljohani, King Abdulaziz University, Saudi Arabia

Mitra Arami, Arab Open University, Kuwait

Khin Mi Mi Aung, Agency for Science, Technology and Research (A*STAR), Data Storage Institute, Singapore

Reinhold Behringer, Leeds Beckett University, United Kingdom

K. Chandrasekaran, National Institute of Technology Karnataka, India

Sidney Chapman, Self-Employed, Australia

Pethuru Raj Chelliah, IBM India, India

Tzu-chun Chen, TU Darmstadt, Germany

Dickson Chiu, The University of Hong Kong, Hong Kong

Darren Chong, Singapore, Singapore

Wendy Currie, Audencia, Nantes, France

Natalia Kushik, Tomsk State University, Russian Federation

Bu Sung Francis Lee, Nanyang Technological University, Singapore

Tope Omitola, University of Southhampton, United Kingdom

Shuqin Ren, Data Storage Institute, Singapore

Jose Simao, Instituto Superior de Engenharia de Lisboa, Portugal

Luis M. Vaquero, HP Labs, United Kingdom

Yun Wan, University of Houston, United States

Junbo Wang, University of Aizu, Japan

Tomasz Wiktor Wlodarczyk, University of Stavanger, Norway

Neil Y. Yen, University of Aizu, Japan

Fan Zhang, MIT, United States

FOREWORD

The Emerging Software as a Service and Analytics– ESaaSA 2015 was organized in conjunction with CLOSER 2015 in Lisbon, Portugal. This workshop is primarily focused on high quality and innovative research papers from different fields related to the most recent developments in Emerging Software as a Service and Analytics.

Authors will have the opportunity to have their work selected for publication in a special issue of RonPub's Open Journal of Big Data (OJBD), International Journal of Organizational and Collective Intelligence (IJOCI) and a short list of papers presented at the workshop venue will be selected for publication of extended and revised versions in a special section of International Journal of Information Management (IJIM).

The conference was also complemented with an invited speaker Dr. Luís Veiga from INESC-ID, Portugal. ESaaSA has invited a number of scholars who have experience of publishing high-quality of papers and has offered open calls for authors from top-tier companies. All the high-quality submissions were reviewed by at least two program committee members who were external to each group of authors.

ESaaSA 2015 received 14 paper submissions. From these, 5 papers were published and presented as full papers and 9 were accepted as short papers.

We would like to thank all the authors who took the time to submit papers to ESaaSA, even though they were not finally accepted. We would also like to express our gratitude for the excellent work done by the Program Committee and the members of the Organization Committee.

Victor Chang
Leeds Beckett University, U.K.

Muthu Ramachandran
Leeds Beckett University, U.K.

Gary Wills
University of Southampton, U.K.

Robert Walters
University of Southampton, U.K.

Verena Kantere
University of Geneva, Switzerland

Chung-Sheng Li
IBM, U.S.A.

CONTENTS

PAPERS

FULL PAPERS

SHORT PAPERS

PAPERS

FULL PAPERS

Quality of Service for Financial Modeling and Prediction as a Service

Victor Chang and Muthu Ramachandran
School of Computing, Creative Technologies and Engineering, Leeds Beckett University, Leeds, U.K.
{v.i.chang, m.ramachandran}@leedsbeckett.ac.uk

Keywords: Quality of Service (QoS) for SaaS, Financial Modeling and Prediction as a Service (FMPaaS) QoS, Performance and Accuracy Test for FMPaaS QoS.

Abstract: This paper describes our proposal for Quality of Service (QoS) for Financial Modeling and Prediction as a Service (FMPaaS), since a majority of papers does not focus on SaaS level. We focus on two factors for delivering successful QoS, which are performance and accuracy for FMPaaS. The design process, theories and models behind the FMPaaS service have been explained. To support our FMPaaS service, two APIs have been developed to improve on performance and accuracy. Two major experiments have been illustrated and results show that each API processing can be completed in 2.12 seconds and 100,000 simulations can be completed in an acceptable period of time. Accuracy tests have been performed while using Facebook as an example. Three points of comparisons between actual and predicted prices have been undertaken. Results support accuracy since results are between 93.72% and 99.63%.

1 INTRODUCTION

The complexity of large scale financial cloud computing services that require high speed and high precision systems grows exponentially. Services of large scale financial cloud computing and grids are enormous in recent years. Some of them are used for weather forecasting, simulation of aircraft and military services, atmospheric and planet study, remote sensing, large scale data analysis, aerospace research, large scale computational fluid dynamic services, aeronautics and automobile industries, and financial simulations. More recently, predication models used by these applications have become increasingly important (Cantor and Royce, 2014). As a result, understanding the behavioral aspects of such systems is important for the design in the quality of service. Some characteristics of large scale financial cloud computing services include:

- High speed and highly parallel
- Real-time
- Virtually connected nodes of systems
- Grid is an infrastructure for large scale financial cloud computing and other resources
- High precision and accuracy

To manage largely-scale software in the cloud, software components and also known as service components are used. The aim is to provide a self-contained entity that can be adapted to the required environment quickly and easily. To elaborate this further, software components design for large scale financial cloud computing and grids have become major issues in recent years and in years to come (Silvestri et al., 2006; Albodour et al., 2012). They have all claimed the importance of software components which will dominate large scale financial cloud computing and grid services. Albodour et al., (2012) propose a model, Business Grid Quality of Service (BEQoS), to measure key metrics and provide added value for commercial and business Grid applications. They use the GridSim software to demonstrate their proof-of-concepts with supporting results to show that reliability and affordability can be achieved. Silvestri et al., (2006) assert that the future large scale financial cloud computing and grid services can be completely built in a bottom-up fashion using software components deployed on various locations and interconnected to form a workflow graph and to re-configure themselves as and when needed during run-time to self manage those services that may in need.

In this paper, we propose a QoS requirements engineering model to assert certain subsets of activities that must be identified and assessed for a large scale financial cloud computing and grid services where the main emphasis has been given to non-functional requirements that match onto the characteristics of such Services. In all the applications and Software as a Service (SaaS), financial applications require on-demand services

that are offered by cloud computing with cost-benefits. Hence, financial domain has begun to reap this benefit with emerging financial SaaS such as FinancialForce developed jointly by SalesForce, NetSuite, Intacct, and Oracle's financial SaaS. According to NetSuite (2014), FinancialForce helped companies increase their revenues by 95%. Accenture (2011) reports on financial technology trends and high performance computing prediction in the following category:

- Leveraging technology to address new & change in regulations
- Reliable and globally harmonized financial systems
- Add value through strategic applications
- Harvest benefits from technology

According to Accenture (2011), SaaS should be simple, efficient, engaging, accessible, clearly structured, intuitive, and supportive. While keeping this set of requirements as design criteria, a SaaS component model and a service architecture should be designed to support flexibility, scalability, and adaptability. This paper has proposed an integrated service-oriented architecture and SaaS component model for financial domains which provides required scalability, flexibility and customization that are at the heart of a financial SaaS.

There are a number of QoS factors that affect quality of a cloud service. We have proposed a set of QoS attributes that are keys to success of cloud services, in particular, Financial Modeling and Prediction as a Service (FMPaaS) where accuracy and performance are the key benefits of such services which has been achieved. To demonstrate accuracy, two types of the accuracy test were given. The first type was focused on the overall accuracy and the second type was focused on three point selection. One example will be illustrated to support accuracy for our FMPaaS.

1.1 QoS for Financial Modeling and Prediction as a Service (FMPaaS)

Cloud is committed to providing everything as a service and QoS can provide multiple parameters that are required by financial cloud computing services. There are a number of QoS metrics to be considered for FMPaaS. In our previous work (Chang, 2014), we demonstrated the use of FMPaaS in business intelligence applications and identified six important factors. The importance of each factor can be measured in the scale between 1 and 10. A complete set of QoS factors that affects FMPaaS are identified in Figure 1 and some which have been validated in our earlier project on FMPaaS (Chang, 2014) and are summarized as follows:

- Usability: Most of QoS APIs are easy to use except one API requires further training. The overall score is 8 because at least 80% of the tools are easy to use and their manuals are self-explanatory. The other 20% of the functionalities require specialized knowledge about financial modeling to compute complex models.
- Performance: Performance on QoS is good. Computation takes a short time to get results. The score is 8.
- Security: QoS needs third party software and is not a model with a high level of security. Basic authentication and authorization can still be achieved. As a result, the score is 4.
- Computational accuracy: Computational QoS results are accurate. Some banks have used QoS to calculate pricing and risks, and are close to the actual values. But QoS requires have accurate input values before getting the final results. This level of dependency is a limitation to prevent it to score 10. The overall score is 8.
- Portability: QoS is highly portable in most of the systems. All operating systems and computational devices can run QoS applications. The overall score is 9.
- Scalability: QoS tools are highly scalable. It can run on a single processor desktop, or clusters of high-end servers. Input variables can be highly adaptable to a wide range of values.

These scores for QoS are based on the results of expert reviews of eleven experts. Follow-up improvements are required to support the QoS model.

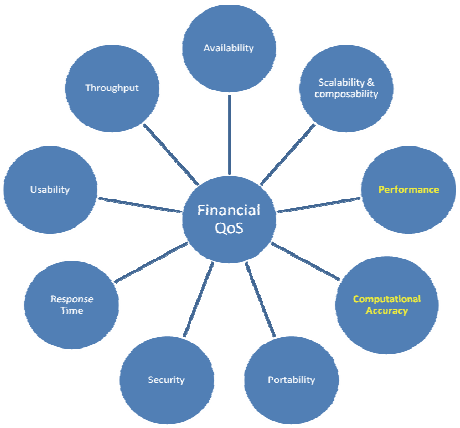

Figure 1: QoS Metrics to Measure.

In addition to these well know parameters to measure QoS, we have also defined a clear model and equation to measure QoS in terms of satisfaction of services on the fly. We highlight important factors essential for QoS success, with more emphasis paid on performance and accuracy. Referring to Figure 1, a list of QoS parameters are used in our work to evaluate service quality. We highlight important factors essential for QoS success, with more emphasis paid on performance and accuracy.

1.2 Our Approach in QoS for Financial Modeling and Prediction as a Service (FMPaaS)

In review of all the six factors influencing QoS, we have already demonstrated the importance of security in our papers (Ramachandran and Chang, 2014). In this paper, we will elaborate on these factors, in particular performance and accuracy. The reasons are as follows. First, literature presented in Section 1.1 does not provide details in accuracy. While SaaS is essential to sectors such as finance and medicine which require an extremely high level of accuracy, any errors or glitch may cause damaging impacts. If FMPaaS calculates incorrect results such as advising investors to buy a particular stock with millions of pounds, or a reliable stock at a particular instance with millions of pounds, they can bear the consequence. This means that the emphasis in QoS accuracy is essential for Cloud Computing.

Second, there is an increased demand to offer accurate predictive services, since the inaccurate results may cause financial loss, loss of company reputation, loss of consumer confidence. This is a type of QoS that have not been presented in the research computing community. For example, if they lose out million of pounds due to the misleading predictive results from similar FMPaaS services, it may result in bankruptcy (Lehman Brothers), loss of reputation (UBS) and loss of investors apart from the direct loss of money. Similarly, simulations related to human bodies such as brain, heart and vital organs are important to determine the most likely scenarios for patients receiving treatments for several years.

With regard to FMPaaS, one of our contributions to QoS is the notion of service satisfaction index which can be in-built as part of a service specification. FMPaaS index allows users evaluate services based on their merits in real scenarios and also supports service reusability, a key benefit of service computing. In reviewing all factors contributing to QoS success, we focus more on accuracy and performance to ensure that our FMPaaS can provide as correct and swift as possible for investors. We emphasize on the software design approach for FMPaaS QoS and use one example to illustrate our proof-of-concepts.

2 INANCIAL MODELING AND PREDICTION AS A SERVICE QoS

This section describes the system design for Financial Modeling and Prediction as a Service (FMPaaS) QoS, which is essential in a few disciplines. For example, e-government applications require open, flexible, interoperable, collaborative and integrated architecture to provide services. These services can be made available as stand alone, integrated, componentized, web based service component, composite service (a set of interconnected services), virtualized services (cloud based), and dynamically re-configurable services. This vision is similar to the Open Group's (2009) Service Integration Maturity Model (OSIMM), which provides:

- A process roadmap for attaining key practices with metrics
- Seven levels of maturity to improve
- A quantitative model for assessing current practices and to improve with recommended practices

As mentioned earlier section, service components are useful to manage system complexity and reuse of services during autonomous service composition. The key challenge is to design a service component that supports service characteristics discussed earlier. A service component can be defined as a self autonomous service which provides two sets of services: provider business services and required business services. The provider business service (often shown with a lollypop notation and the naming convention starts with I) is a set of services offered to other services to compose where as the required business services (often shown as a semi-arc notation) are a set of services that are required by this service in order to compose successfully. In this work, we have proposed a component model for FMPaaS applications as shown in Figure 2, which the required services include Income statement, ICashFlow statement, Ie-taxation, IFSA regulations. IFSA provides interface service integration for Financial Authority regulations. Ay investment

service providers can integrate their work to this FMPaaS service component model, which is adaptable to regular updates in regulations. By doing so, FMPaaS can provide scalability and flexibility for financial analysts. These services can be made available as stand alone, integrated, componentized, web based service component, composite service (a set of interconnected services), virtualized services (cloud based), and dynamically re-configurable services.

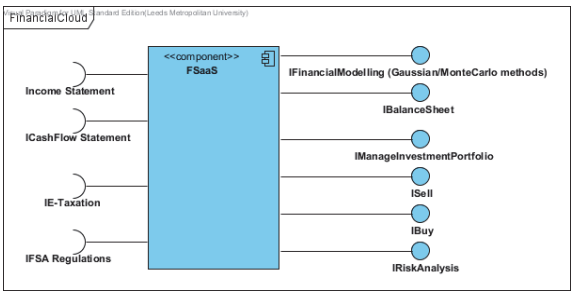

Figure 2: FMPaaS Service Component Model.

The next step in the design process is to design service-oriented cloud architecture for FMPaaS where all aspects of the corporate financial service are integrated and composed based a set of SLA and governance. The architecture presented in this paper is based on a critical review and analysis of a number of existing architectures for FMPaaS applications. Further to this, the SOA based architecture consists of four distinct levels of abstraction layers which are connected and communicated by messages through a core communication channel known as a service bus or a central bus. These layers are: 1) a business layer with a dedicated set of services; 2) an orchestration layer with a set of services where new services can be composed; 3) an FMPaaS layer that supports integration of services, government departments and local governments, and 4) an e-business layer that supports new businesses and integration of data. The SOA based architecture for FMPaaS services, then ensures that it achieves the expected service-oriented design factors such as customization, cost-effectiveness, availability, etc. The service-oriented FMPaaS architecture is shown in Figure 3.

Referring to Figure 3, at the business and orchestration layers provide high level service composition based on new business perspective and policies (both political and economical factors). Mostly, the customization and the new business needs arise from these two key variables. The sub-systems such as registration control, security control, integrated services for FMPaaS applications control, and communications channels help to achieve customization at a higher level of abstraction without affecting underlying business logic services. These are communicated and connected to layers below using a concept of service bus known as FMPaaS secured service bus. The layer below the business layer provides services for various FMPaaS departments, and external suppliers (E-Business layer). Software components for large scale financial cloud computing services require a detailed analysis of the domain and its boundary in order to define a collection of components for large scale financial cloud computing services that are highly reusable and scalable. A good SaaS design should introduce a domain analysis process which allows us to define a set of common definitions, domain classification, domain boundaries, domain models, design artifacts, and design guidelines that are based on those domain criteria.

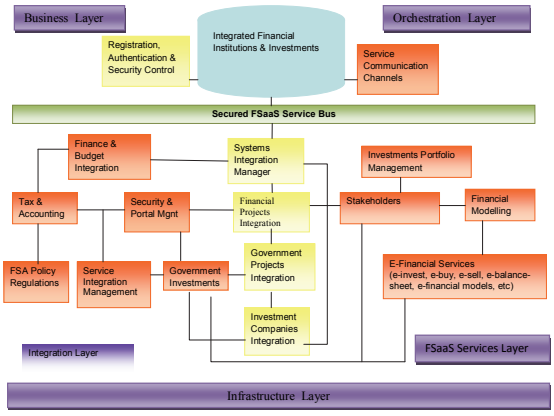

Figure 3: Service-oriented Architecture for FMPaaS.

3 MODELS AND THEORIES BEHIND FMPaaS

The current work on QoS (Lee et al, 2009; Mukhopadhyay, 2012; Shehu et al., 2014) have proposed a number of frameworks and are useful in its own merits. However, they only have an emphasis on other non-functional attributes and then claim non-functional attributes as QoS parameters. Similar to Albodour et al., (2012), our proposed model is to provide commercial uses for research institutes, financials services and general public who are involved or interested in stock market analysis. The main difference between our work and Albodour et al. (2012) is that we use our own development of work. We have developed a comprehensive approach based on the development

of FMPaaS extended from our current work, which aims to distinguish QoS attributes clearly; helps to identify them from requirements to model financial cloud and then validate services against those attributes. These include the followings:

1. Based on the reputable models – the chosen model is the Heston Model (which includes the Wiener process and the Stochastic Volatility) and the Visualization APIs to compute the best pricing and risks for different scenarios.
2. Accuracy to compute and track volatility – FMPaaS can track the movement of volatility and help investors make a better judgment for investment when prices are high and volatility is low. Our FMPaaS can compute pricing and risk values to several decimal places and also calculate its mean, lower and upper range to get our results as accurate as possible.
3. Performance – all calculations should be completed within seconds to ensure all services can be delivered in an acceptable time frame.

3.1 Models Used for FMPaaS

Models behind FMPaaS are essential for the calculation, processing and presentation of financial computation in the Cloud. Our previous work explains all the associated models, including the choice of the models, their associated formulas, how they can be used in the development of FMPaaS. In summary, models include (Chang, 2014):

1. Heston Model
2. Wiener Process
3. CIR (Cox, Ingersoll and Ross) Model
4. Runge–Kutta method (RKM)

The use of all the models for FMPaaS can match accuracy and optimize the performance. The summary of their descriptions is as follows.

3.1.1 The Heston Model

The Heston Model has a close relationship with Black-Scholes model, since it relaxes the constant volatility assumption in the classical Black-Scholes model by incorporating an instantaneous short term variance process (Albrecher et al., 2006). In other words, the Heston Model can be used in a more flexible way and is not as theoretical-oriented as the classical Black-Scholes model does. In addition, there are both the Wiener process and the CIR process related to the Heston Model. Heston Model has been explained in our previous work and it can still be very useful for undertaking business intelligence services and prediction of financial modeling (Chang, 2014).

3.1.2 The Heston Model

The Wiener process is a stochastic process with independent and stationary increments, which means the motion of a point whose consecutive displacements are independent and random with each other. The Wiener process has Lévy characterization has continuous martingale with W0 = 0 and quadratic variation [Wt, Wt] = t. This implies that Wt2−t is a martingale (Cox et al., 1985; Kloeden and Platen, 1999). The basic Heston model assumes that S_t, the price of the asset, is determined by a stochastic process (Cox et al., 1985; Kloeden and Platen, 1999). The Heston Model has a CIR process involved, which is a Markov process with continuous paths defined by the following stochastic differential equation (SDE). The variable include Wiener process (i.e., random walks) with correlation ρ dt. The parameters in the Heston model for providing input in the computation in Section 4 represent the following:

- μ is the rate of return of the asset.
- θ is the long variance, or long run average price variance; as t tends to infinity, the expected value of ν_t tends to θ.
- κ is the rate at which ν_t reverts to θ.
- ξ is the volatility of the volatility; as the name suggests, this determines the variance of ν_t.

3.1.3 The CIR Model

The CIR process is used to model stochastic volatility in the Heston model, which aims to resolve a shortcoming of the Black–Scholes model which corresponds to the fact that the implied volatility does tend to vary with respect to strike price and expiry. By assuming that the volatility of the underlying price is a stochastic process rather than a constant, stochastic volatility can make it possible to model derivatives more accurately (Cox et al., 1985; Wilmott and Wilmott, 2006).

3.1.4 The Runge-Kutta Method

The Runge–Kutta method (RKM) is a technique for the approximate numerical solution of a stochastic differential equation (SDE) (Hull and White, 1987; Wilmott, 2006). RKM can be used to generalize the ordinary differential equation to SDE. To elaborate further, the Ito diffusion X satisfying the following Ito stochastic differential equation (Hull and White,

1987; Wilmott and Wilmott, 2006). Details of formulations can be referred to Chang (2014).

3.2 Methods for FMPaaS Calibration

This section describes methods for FMPaaS calibration, which is used in a way that a known observation of the dependent variables is used to predict a corresponding explanatory variable. The root-mean square error (RMSE) and Moving Window (MW) are identified as the methods to perform FMPaaS calibration.

3.2.1 The Root-Mean Square Error

The Root-Mean Square Error (RMSE) is used to measure of the differences between values predicted by a model or an estimator and the values actually observed. RMSE also determines the goodness of fit of the Heston Model presented by Cox et al. (1985) and Hull and White (1987).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (X_{obs,i} - X_{model,i})^2}{n}} \qquad (1)$$

where n is the number of quoted options, X_{obs} is observed values and X_{model} is modelled values at time/place i. The parameters required for RMSE include (ν_0, κ, θ, ξ , ρ) used for calibration and ν_0 is the instantaneous variance at the starting point. Referring to formula (2), the rate of return of the asset can be calculated by multiplying κ and difference between θ and ν_0.

3.2.2 The Moving Window

The Moving Window (MW) estimate is a suitable model in the use of VIX options, which are provided daily to track market values of volatility. MV can be computed as the mean of variance of the stock price process over the time series window that moves forward in time. MW is used to compute the forecasted movement in the Heston Model.

3.2.3 Average Absolute Percentage Error (APE) and Aggregated Relative Percentage Error (ARPE)

The average absolute percentage error (APE) of the mean price and aggregated relative percentage error (ARPE) are additional formulas for calibration to construct the best fit in financial computation, and thus improves the accuracy and performance of the calculations (Wilmott, 2006; Kloeden, and Platen, 2012; Guillaume and Schoutens, 2012). A limitation with APE is that it may cause a problem. A few of the series with a very high APE might distort a comparison between the average APE of time series fitted with one method compared to the average APE when using another method. To overcome this limitation, another model, aggregated relative percentage error (ARPE) is used.

3.3 Services on Offer

This section explains two types of services on offer for FMPaaS QoS. The architecture adopts the private cloud at the University of London Computing Centre (ULCC) data center and Southampton clusters, where the processing took place in ULCC. Two types of services are as follows.

- Heston Volatility and Pricing as a Service (HVPaaS): The request started and completed at Southampton clusters, including the processing of the HVPaaS. The objective is to track volatility and pricing simultaneously since both can change significantly during the volatile period. The metrics are provided by the respective inputs of Heston model except volatility, which is provided by VIX.
- Business Analytics as a Service (BAaaS): After analyzing the numerical computation of volatility and pricing, the next step is to compute them as a Business Analytic. This makes the analysis much easier and the stakeholders can understand. After the processing of HVPaaS completed in Southampton, results are sent to ULCC in London, where both sites can process BAaaS. This service is regarded as the case of a complete FMPaaS QoS.

Application Programming Interfaces (APIs) are used to illustrate how to use these two services. In BAaaS, it has two APIs as follows.

1. **FinancialData** API – this allows the BIaaS Cloud to obtain financial data from Google Finance and have all the major stock market data, particularly the US and UK stock exchange data.
2. **TradingChart** API – this allows the financial data to be presented in the trading chart format similar to the visualization services offered by London Stock Exchange and Thomson Reuters. Additional functions can allow analysts to use the MW model to compute forecasted movement. "TradingChart" is the API to demonstrate both models (Heston and Financial

data) can work together to deliver an integrated service. Results of the experiments will be presented in Section 4.

3.4 Measurement of FMPaaS QoS

This section describes the measurement of FMPaaS QoS, which aims to demonstrate the significance of performance and accuracy. In terms of performance, the execution time for all APIs should be recorded to check their completion time is within seconds. Experiments involved with multi-core and multi-node processing are included to illustrate the performance issue. To demonstrate accuracy, an approach is to compare the predicted result from the FMPaaS QoS with the actual results generated by the market such as the New York Stock Exchange or London Stock Exchange. The end results of these APIs, particularly the TradingChart API (the last one of all FMPaaS services), can correspond to the predicted results of the FMPaaS analysis. The actual results can be imported directly from Google Finance. The difference between the actual and predicted results can correspond to the percentage of accuracy. The objective is to maintain all differences within 5% difference to ensure a high quality of accuracy to be achieved.

4 ACCURACY TESTS AND RESULTS OF PERFORMING FMPaaS QoS SIMULATIONS

This section describes the accuracy tests of the selected stocks listed on the New York Stock Exchange. Some of these selected stocks are the continuation of our previous study which analyzed stocks between mid-May 2012 and early July 2013. Hence, we will analyze the stocks between early July 2013 and mid-May 2014. Additionally, some of the new selected stocks such as Citi and GE are used to analyze the accuracy of FMPaaS results. Our previous work has shown the stocks of Facebook, Apple, IBM and Microsoft between mid-May 2012 and end of June 2013 and these four stocks are used again for FMPaaS analysis.

4.1 The Overview of the FMPaaS

This section presents the overview of the FMPaaS, including the end results of the analysis shown in Figure 4. The first section of Figure 4 is the main area of FMPaaS QoS, where the y-axis shows the price and the x-axis shows the time scale. There are upper and lower lines, which are predicted indexes based on the stock values every ten minutes ago. As explained in our previous work, both upper and lower limits offer 95% of confidence interval (CI) for the predictive modeling. The purple line in the middle is the baseline based on the prediction. The blue line in the middle is the predicted value line based on the values given 10 minutes ago and without using the 95% CI approach. The second section represents the trading volume. The third section represents the relative strength index, which means how active the stock movement is compared to 50 as the baseline. In this case, we are only concerned about the first section, the accuracy and performance of the actual and predicted index movements.

Figure 4: The full FMPaaS result showing Facebook stock prices, volume and relative strength between 2 July, 2013 and 16 May 2014

4.2 Performance Test: The Experiments with APIs

As explained in Section 3.3, development of APIs is essential for FMPaaS to measure the effectiveness of QoS. Our previous work also demonstrates the use of two APIs, “FinancialData” and “TradingChart”, which display the outputs of FMPaaS based on the calculation and computation of formulas presented in Section 3. The outputs measure the following two items:

- The status of the return, which are the prices of the assets at the times that sales are intended;
- Volatility, which represent the variable market risk associated with the sale or buy activities.

Experiments with these two APIs are important

since they will determine the performance of generating results and accuracy of the results received. To present the results of experiments, the hardware specifications are described in Section 4.2.1. Steps and processes involved with two experiments are then presented in Section 4.2.2 and 4.2.3 respectively.

4.2.1 Infrastructure Used for Experiments

University of London Computer Center (ULCC) was used for the experiments. ULCC has advanced Cloud and parallel computing infrastructure and network attached storage (NAS) service. It has CPUs totalling 30 GHz, 60 GB of RAM and 12 TB of disk space for experiments. Fiber optic network offering the 10 Gb network speed was used for experiments. The network was connected to the first private clouds based at Greenwich, which has a total of 9 GHz CPU and 20 GB RAM. The infrastructure at ULCC is also connected to the second private cloud based at the University of Southampton, which have 6.0 GHz and 16 GB RAM in place. There is the third private cloud based at the author's venue at Southampton, which has the capability is 24.2 GHz CPU and 32 GB RAM. All the three private clouds located in Greenwich and two places at Southampton have been connected to ULCC through the fast fiber optic networking and the VMWare infrastructure. Before experiments took place, preliminary work had been tested and all the outputs could be successfully computed. The distance between different private clouds did not make a difference in the execution time during the preliminary phase of the experiments.

4.2.2 Execution Time for a Single API Processing

This section presents results of processing each API in two settings. The first experiment was undertaken between the two private clouds at Southampton. The second experiment was undertaken while utilizing both the Southampton and ULCC clouds. In other words, results should be sent to ULCC for processing and returned back to Southampton. The execution time is the total time of processing mathematical modeling on the APIs on the server and response time to the client. The first experiment was expected to take less time due to the shorter distance. All experiments were conducted five times with the mean values taken as the execution time and the standard deviation was the difference between the highest and lowest values. The results of API experiments were presented in Table 1.

Table 1: The execution time for each API or process in the local environment ($p < 0.005$).

API or process	Southampton execution time (sec) and standard deviations	ULCC: execution time (sec) and standard deviations
FinancialData	2.04 (0.10)	2.12 (0.12)
TradingChart	1.11 (0.03)	1.19 (0.06)

4.2.3 Execution Time for 100,000 Simulations of API Processing

Results in Section 4.2.2 show the average execution time of one simulation per API processing. To test the performance, the large-scale simulations are required (Guillaume and Schoutens, 2012). Our FMPaaS can offer up to 100,000 simulations per service to test the scenarios that if there are 100,000 service requests happen every second, whether our FMPaaS can still provide services smoothly without degrading the service. The aim of this experiment is to demonstrate that our FMPaaS can support 100,000 service requests and achieve a good execution time. Availability was 100% at the time that those experiments were taken, with the network and VMs working in excellent conditions. All the experiments were taken five times with the mean values taken as the execution time and the standard deviation was the difference between the highest and lowest values. Results are presented in Table 2. 100,000 simulations on the API could be completed in 200,645 seconds, or 55 hours, 44 minutes and 5 seconds.

Table 2: The execution time for 100,000 simulations of API processing in the ULCC ($p < 0.005$).

API or process	Southampton execution time (sec) and standard deviations	ULCC: execution time (sec) and standard deviations
FinancialData	200432 (488)	200645 (499)
TradingChart	110135 (417)	110348 (429)

All the standard deviations are below 0.5% of the average execution time for all six APIs. The aim for this experiment is to demonstrate that in the event of having 100,000 requests from users in real-time, how the FMPaaS can respond to all the processing. Results also show that FMPaaS can cope with 100,000 requests.

4.3 Accuracy Test

This section describes the accuracy test by using Facebook as an example to illustrate. The focus is to

demonstrate accuracy and performance of using FMPaaS analysis. The execution time of performing this FMPaaS test is 3.15 seconds, which correspond to the sum of processing "FinancialData" and "Tradingchart" APIs. We identify three major points where the predicted asset prices would be compared directly with the actual prices. The reason was that since price values could change all the times, identifying the points for comparison was useful. Additionally, this can ensure prediction to be more focused on the end of the trading activities since they could receive more investors' attention.

Two types of accuracy tests are presented. The first test is focused on the overall level of accuracy, whether all the actual values fall into the upper and lower predicted values within the range of 95% confidence interval (CI). The second test is based on three selection points where the trading activities are at the end of the quarterly business review, or at three obvious points in the FMPaaS result. In Figure 5, points 1, 2 and 3 are chosen due to the location of these points to be checked and noticed easily.

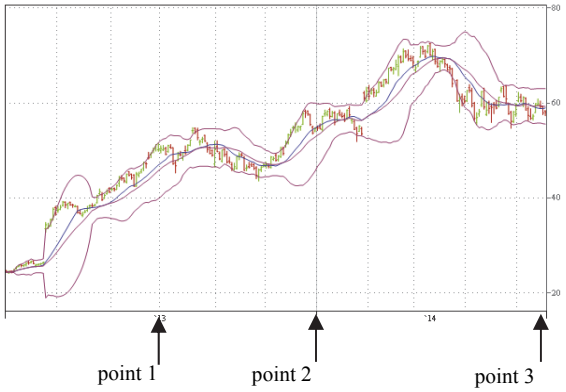

Figure 5: The FMPaaS result showing Facebook stock prices between 2 July, 2013 and 16 May 2014.

Table 3 shows the results of the overall accuracy test. We count the number of datapoints falling outside the 95% CI divided by the total number of datapoints. The results show that about 97% of the actual datapoints, or actual values of Facebook index movements, fall within the 95% CI predictive range. Among those 3% falling outside the predicted range, there is one spot with a red arrow. This happened because Facebook was reported to have more profits than their analysts' forecasted results. However, the market had the mixed reactions in the first few days, which resulted in numerous selling and buying activities. Those who bought thought that Facebook would have a better value at some point. Those who sold thought that it was a time to get their investment back. This explains why our forecasted values slightly deviate from the actual values. Additional calibration can be used to compute the forecast price values and volatility for the three points, where the results can then be used to compare with the actual values for the accuracy.

Table 3: The test of the overall accuracy for Facebook.

Items	Falling within 95% CI lines	Percentage falling outside 95% CI lines	Significant spots falling outside 95% CI lines
Actual values	Yes. 97% of actual values are within the range.	About 3%	Profits were more than their predicted results between 2013/2014 forecast.

To determine the accuracy test, asset prices of the predicted values (input values by Heston model and VIX and computed by models in Section 3) are directly compared with the actual values. See Table 4 for results. Asset prices computed by the predicted value are close to their respective actual values in points 1, 2 and 3, ranging between 93.72% and 99.63% accuracy. Points 2 and 3 have extremely high accuracy and point 1 has an acceptable level of accuracy. The likely reason is that the asset price prior reaching point 1 was on the way up to one and a half months and it was less predictable to forecast the asset price values on the way up in point 1.

Table 4: The test of the three selection point accuracy for Facebook.

Items	Actual value	Predicted value
Point 1	**Asset price = 50.15;** volatility = 1.20; implied volatility = 0.45; time = 0.3	**Asset price = 47.00;** volatility = 1.20; implied volatility = 0.45; time = 0.3. **93.72%** same as the actual value
Point 2	**Asset price = 53.30;** volatility = 0.5; implied volatility = 0.45; time = 0.6	**Asset price = 53.70;** volatility = 0.5; implied volatility = 0.45; time = 0.6. **99.26%** close to actual value
Point 3	**Asset price = 59.01;** volatility = 0.5; implied volatility = 0.35; time = 1.15	**Asset price = 59.23;** volatility = 0.5; implied volatility = 0.35; time = 1.15. **99.63%** same as the actual value

4.4 Discussion

The benefits of adopting FMPaaS are as follows. First, FMPaaS have focused on improving the accuracy for the financial modeling and prediction as demonstrated in the test results. This can also provide new and alternative services for forecasting and investment analysis. Second, FMPaaS can

provide positive impact to the stakeholders and potential investors to understand the market performance, volatility, trading volume and likely predicted movements of their chosen stocks. These two aspects of contributions will help the stakeholders, potential investors and research community to understand the market much better. The benefits offered by FMPaaS are relevant to the themes of Emerging Software as a Service and Analytics to allow the community to know an improved and better Cloud SaaS services being validated and illustrated with reported contributions. The next phase of challenges is to improve the overall level of accuracy from 95% to 98% and above; improve the point accuracy as close as to 99.99% and raise three points of evaluation and testing to six points to ensure there is a greater coverage of accuracy tests.

5 CONCLUSION AND FUTURE WORK

A large number of QoS papers focus on the hardware infrastructure and Service Level Agreement with the lack of explanation and further development for SaaS. We explain the motivation and significance of QoS for FMPaaS, which is our main service for finance and business intelligence. Six factors for delivering FMPaaS QoS have been illustrated, where the emphasis for this paper is on performance and accuracy. We first start with the design process and methodology for FMPaaS, and then explain the theories behind FMPaaS. APIs are provided in the FMPaaS, where "FinancialData" and "TradingChart" are the two APIs that have been developed and then used in the experiments for performance tests. Two types of experiments were conducted. First, each API was tested five times top get the mean execution time to generate outputs. All execution time was completed within 2.12 seconds. Second, large scale of 100,000 simulations was performed to test whether APIs can provide real-time services. Results show that 100,000 simulations on the API could be completed in 200,645 seconds, or 55 hours, 44 minutes and 5 seconds with a low percentage of standard deviations. Accuracy had been conducted to test the differences between the predicted and actual values. Three points of comparisons for Facebook stock were used for accuracy test since they represented the end of all transaction activities. Results show that accuracy tests had between 93.72% and 99.72% of accuracy while comparing the actual and predicted values of the asset prices of Facebook stock. Our future work will include the improvement of our performance and accuracy tests. We will also use more companies to illustrate that our FMPaaS can provide better services and accuracy while comparing the actual and predicted values of asset prices.

REFERENCES

Accenture, 2011, Accenture Financial Trends slides, http://www.slideshare.net/fullscreen/ramblingman/accenture-financial-saa-s-external-presentation-final/3, accessed on April 2014.

Albodour, R., James, A., N. Yaacob, 2012, High level QoS-driven model for grid applications in a simulated environment. *Future Generation Computer Systems, 28(7), 1133-1144*.

Albrecher, H., Mayer, P., Schoutens, W., and Tistaert, J., 2006, *The Little Heston Trap, Technical paper, September*.

Cantor, M. and Royce, W., 2014, Economic Governance of Software Delivery, *IEEE Software, 31(1)*.

Chang, V., 2014. The business intelligence as a service in the cloud. *Future Generation Computer Systems*, *37*, 512-534.

Cox, J.C., Ingersoll J.E. & Ross, S.A. 1985, A Theory of the Term Structure of Interest Rates, *Econometrica 53: 385-408*.

Durrett, R., 2000, Probability: theory and examples, 4th edition. *Cambridge University Press, ISBN 0-521-76539-0*.

Guillaume F., and Schoutens, W., 2012, Calibration risk: Illustrating the impact of calibration risk under the Heston model, *Review of Derivatives Research, 15:57–79*.

Hull, J., and White, A., 1987, The Pricing of Options on Assets with Stochastic Volatilities, *The Journal of Finance, 42(2)*.

Lee, J. Y., Lee, J. W., Cheun D. W. & Kim S. D., 2009, QoS A Quality Model for Evaluating Software-as-a-Service in Cloud Computing, *the Seventh ACIS International Conference on Software Engineering Research, Management and Applications*.

Kloeden, P.E, Platen, E., 1999, Numerical Solution of Stochastic Differential Equations. *Berlin: Springer. ISBN 3-540-54062-8*.

Mukhopadhyay, D., Chathly, F. J., Jadhav, N. N., 2012, QoS Based Framework for Effective Web Services in Cloud Computing, *Journal of Software Engineering and Applications, 5, 952-960*.

NetSuite, 2014, white paper and software, product http://www.netsuite.co.uk/portal/uk/seo-landing-page/accounting-2/main.shtml?gclid=CLK9k5q-37sCFTHLtAodikoAzw, accessed on April.

Open Group, OSIMM, 2009, from https://www2.opengroup.org/ogsys/jsp/publications/PublicationDetails.jsp?publicationid=12450, Retrieved Oct 2013.

Ramachandran, M., Chang, V., 2014 Cloud Security proposed and demonstrated by Cloud Computing Adoption Framework, *the first international workshop on Emerging Software as a Service and Analytics, Barcelona, Spain, 03 - 05 April*.

Schulze, B., Coulson, G., Nandkumar, R., Henderson, R., 2006, Special Issue: Middleware for Grid Computing: A 'possible future', *Concurrency and computation: practice and experience, 10.1002/cpe.1132, Wiley*.

Shehu, U., Epiphaniou, G., Safdar, G. A., 2014, A Survey of QoS-aware Web Service Composition Techniques, *International Journal of Computer Applications (0975 – 8887), 89(2), March*.

Silvestri, F. et al., 2006, Toward a search architecture for software components, *Journal of Concurrency and Computation: Practice and Experience: 18:1317-1331*.

Wilmott, P., 2006, Paul Wilmott on quantitative finance, *Wiley (2nd ed.), ISBN 0470018704*.

Scalable QoE Prediction for Service Composition

Natalia Kushik and Nina Yevtushenko
Department of Information Technologies, Tomsk State University, Tomsk, Russia
ngkushik@gmail.com, yevtushenko@sibmail.com

Keywords: Service (Composition), QoE Estimation/Prediction, Logic Network/Circuit.

Abstract: In this paper, we present an approach for scalable QoE estimation/prediction of a composition of given services. The approach relies on using logic circuits/networks for the QoE prediction. Given two logic circuits that predict the QoE values of two service components, we propose a method for synthesizing the resulting logic circuit that predicts the QoE of the overall service composition. As the complexity of this resulting circuit significantly depends on the complexity of an implementation of a MIN function, we present an experimental evaluation of the complexity of the corresponding circuit.

1 INTRODUCTION

The number of services designed for various purposes increases rapidly, and almost all of them are developed for improving or simplifying human life. As an example of the service one can consider a multimedia service, that delivers some video/audio traffic to an end-user, or a web service that allows to book a hotel or to buy some products online, etc. As all these services are developed "for people", the Quality of Experience metrics (QoE) remains the most common metrics to evaluate their quality.

The QoE is used to measure the end-user satisfaction with a given service and thus, the problem of its evaluation remains one of the most challenging problems in the artificial intelligence area. The reason is that in order to evaluate the QoE, it is necessary to 'guess' how much an end-user would like or dislike a given service. This problem is often solved with the use of various self-adaptive models that can accept service parameter values as inputs and return the QoE value as an output. If a model behaves in a wrong way for some newly emerged input/output pairs, the model can be trained by itself or by an external 'teacher' that could be a service provider. Most popular self-adaptive models are decision trees (see, for example, Mitchell, 1997; Pokhrel, J., Mallouli, W., and Montes de Oca, E., 2013), neural networks (Ahmed et al., 2012; Al-Masri and Mahmoud Qusay, 2009), fuzzy logic formulae (Lin et al., 2005; Torres et al., 2011), and logic circuits (Kushik et al., 2014). All these models have their own advantages, as well as the known drawbacks. Most common criteria that a researcher or a service provider should take into account are the QoE prediction ability of the model and the scalability of the "teaching" process. It has been previously shown that the approach proposed by Kushik et al. in 2014 allows to adequately predict the end-user satisfaction with a given service, and, at the same time, to perform the model adaptation in a scalable way (Kushik et al., 2014). This approach is based on logic networks, in particular, combinational circuits, for the QoE prediction. The initial logic network is derived based on statistical data that are gathered from experts, developers and/or end-users who agreed to provide a feedback about the service quality. The circuit accepts the service parameter values encoded as Boolean vectors and outputs the Boolean vector that corresponds to the encoded QoE value. The circuit is a self-learning machine, i.e., when new statistical data appear the circuit is checked for having the corresponding behaviour and if the behaviour does not correspond to newly emerged data the circuit is resynthesized. Such resynthesis can be efficiently performed using various tools (see, for example, Berkeley Logic Synthesis and Verification Group, ABC).

Once the QoE of a given circuit is carefully estimated, one can use this service not only as a single self-sufficient entity, but also as a part of a 'big' service composition. In this case, the problem arises of predicting the QoE of this composition. It is well known, that even if the service components have the high QoE value for a given statistical

pattern, the QoE of the composition is not necessarily high for this pattern. Therefore, the composition QoE value has to be effectively predicted. Given two service components, and two logic circuits for predicting component QoE values, we propose a technique how to synthesize the resulting logic circuit that predicts the QoE of the service composition. The technique relies on scalable operations over logic networks, such as introducing additional inputs and connecting nodes in the circuit to combine particular circuit parts. We introduce a special circuit implementation of the *minimum* function that outputs a minimal integer of two integers. This circuit is further used as a part of the resulting logic network that predicts the QoE value of the service composition. The algorithm provided in the paper takes into account the fact that the user satisfaction can be only decreased in the service composition. The reason is that if a user is not satisfied with a given service component, his/her satisfaction cannot be increased with the use of the other components, i.e., our approach assumes the worst-case scenario. We notice, that this scenario supports the scalability of the approach, since we are not interested in the composition details, i.e., compositional patterns, differently from predicting some objective service parameter values (see for example, Zheng et al., 2013). Furthermore, we discuss how the proposed QoE estimation technique can be adapted to the case when the composition QoE is calculated not as the minimum function but as more complex mathematical formula.

Therefore, the main contribution of this paper is an approach for estimating the QoE of the service composition, when the QoE of each service component is calculated by a corresponding logic circuit. We also provide the preliminary experimental evaluation for a proposed approach addressing the complexity of parts of the resulting circuit. These experimental results clearly show the approach scalability.

The rest of the paper is organized as follows. Section 2 contains the preliminaries. A running example for a service composition and its QoE prediction is given in Section 3. A scalable approach for estimating the QoE of the service composition as well as the experimental evaluation of the complexity of the overall circuit are given in Section 4. A discussion on possible extensions of the proposed approach is presented in Section 5. Section 6 concludes the paper.

2 PRELIMINARIES

In our normal human life, we are surrounded by *services*. Those can be *web services* that represent specific software designed to support interoperable machine-to-machine interaction over a network (Booth et al., 2004) or *multimedia services* that are used to deliver a multimedia traffic to an end-user (Pokhrel, J., Wehbi, B., Morais, A., Cavalli, A., and Allilaire, E., 2013). One can consider other types of services, not directly related to Computer Sciences area, such as cleaning service, delivery service, booking service, etc. Anyway, all these services are developed to improve or to simplify the human life quality and thus, not a single service is left without evaluating the quality of this service. There exist various metrics to evaluate the service quality where the most known seems to be the Quality of Service (QoS) metrics. The QoS can be defined as a vector with components which are values of given attributes (parameters) that can be objectively measured (Kondratyeva et al. 2013). We mention that there have been performed a lot of research and some interesting contributions have been made regarding the estimation of the QoS for a composite service (El Hadad et al., 2010; Zheng et al., 2013).

However, the most interesting metrics to estimate the service quality is the Quality of Experience (QoE) that represents a user satisfaction (see, for example, Winckler et al., 2013). In spite of the fact that the QoE is more difficult to evaluate, this metrics is more close to the adequate description of the service quality, since the main purpose of each service is to satisfy an end-user. In other words, the algorithm for the QoE evaluation has to be adapted to a human's brain in order to 'predict' what a user likes/dislikes. That is the reason why different self-adaptive models and algorithms are now used for this purpose. The advantage of a self-adaptive model is that it can be learnt or trained by a 'teacher' or by itself according to the feedback from people who use the service. As usual, an initial model/machine is derived based on some statistical data that contain a number of user/expert opinions about the service. Afterwards, the model can 'predict' the user satisfaction of the service for the given values of service parameters. The more statistical data are gathered the better is the 'prediction'. Moreover, as the model is self-adaptive, when new statistical data appear for which the model does not behave in an appropriate way, the model is adjusted to these new data and this process is the *model training*.

Various self-adaptive models can be used for the

QoE prediction of the service. One of short surveys of these models can be found in (Kondratyeva et al. 2013). In particular, Kondratyeva et al. discuss three most popular self-adaptive models that are used to predict the QoE value for web services. We briefly sketch this survey to provide an overview of the use of self-adaptive models for the QoE prediction. Almost all self-adaptive models rely on pre/post conditions that can be expressed in terms of IF-THEN operator. The first group of machine learning algorithms is based on a Decision Tree (Mitchell, 1997; Pokhrel, J., Mallouli, W., and Montes de Oca, E., 2013) that can be described for a web service as a tree which nodes correspond to service parameters (attributes) while edges are marked with different parameter values (scores). Each tree level corresponds to a single service parameter which can be evaluated by scores that label outgoing edges. The leaves of the tree correspond to different values of the user satisfaction. The decision tree can be derived based on IF-THEN conditions where a path labelling each branch of the tree to a node with a given QoE value corresponds to the conjunction of conditions under IF operator. The decision tree can be learnt based on deriving IF-THEN conditions by adding additional paths. As usual, such pre/post conditions are derived based on experimental results or following some expert opinions. The decision tree provides an algorithm for evaluating the user satisfaction if and only if it is completely specified. Those paths in the tree that are not specified by the conditions have to be somehow augmented in order to predict the user satisfaction in this undefined situation. Thus, the purpose of specifying undefined paths is to "guess" what a user would like or dislike under appropriate conditions. The complexity of the completely specified tree is exponential w.r.t. the number of quality parameters. Other self-adaptive models, such as neural networks and fuzzy logic formulae are known to be more compact. Neural networks are widely used for solving various problems in the artificial intelligent area. Such networks are used in the "machine learning sense" and all the neurons of the network are assumed to be artificial and can be modified by a "teacher" in a given way. Neurons are connected to each other and these connections also can be trained. Usually neural networks without feedbacks are considered and in this case, the network can be divided into levels. Usually, for each neuron there exists a formula that calculates its output according to weighted inputs that is used when coming to the next level via weighted edges. A neural network can accept values of input (QoS/QoE) parameters and depending on the neuron definition and on the weight of distributed connections the network produces the output (the QoE value) (Al-Masri and Mahmoud Qusay, 2009) by changing states from level to level. At the initial step, the network connections are set based on the initial statistical data, i.e., on the set of given input/output pairs. A network *learning* process consists of modifying weighted connections (or a set of nodes) of the network based on new knowledge (more statistical data, for example). In other words, when new statistical data appear the network can be learnt how to modify its connections and possibly, nodes in order to have the correct behaviour. A good alternative to artificial neural networks is a fuzzy logic that was introduced by Lotfi A. Zadeh (Zadeh, 1965) in 1965 and can be also considered for modelling a human behaviour. Similar to a decision tree, a fuzzy model can be built based on a set of IF-THEN conditions that can be combined taking into account how disjunction and conjunction are defined for fuzzy sets. The fuzzy logic model can be learnt by changing *membership* degree of each parameter to the service, i.e., the weight of linguistic values for quality parameters in the resulting fuzzy formula, as well as by changing the relative importance of each quality parameter.

In 2014, Kushik et al. have proposed another self-adaptive model that can be used to predict the QoE value with a given service. Moreover, the proposed approach has been compared with the one, based on using fuzzy logic formulae, and the former has shown the higher scalability (Kushik et al., 2014). This approach is based on analyzing and training of logic networks/circuits that can be effectively performed using the tools developed for logic synthesis and verification. In this paper, we extend the approach proposed by Kushik et al. to the case when a service under investigation is a composition of 'smaller' services, such that the corresponding logic circuits for the service components are known in advance. Furthermore, we address the methods for deriving such logic circuits for various service types and propose a technique for the efficient QoE estimation for a composite service using the same logic synthesis 'apparatus'. That is the reason why we further briefly sketch the necessary definitions related to the logic synthesis. We mention that these definitions are mostly taken from (Kushik et al., 2014).

Definition 1. A *logic network* (circuit) consists of logic gates. Each logic gate has input (-s) and a single output. Outputs of some gates are connected to inputs of the others. The inputs of some gates that are not connected to any other gate output are

claimed to be primary inputs while the outputs of some gates are claimed as primary outputs. In this paper, we consider combinational circuits, i.e., feedback-free circuits which have no latches.

Each gate implements a Boolean function. Most common 2-input gates are AND/OR/XOR/NAND/NOR/XNOR that implement conjunction/disjunction/xor and their inversions. There are also 1-input gates such as NOT/BUFF that implement the inversion and the equality function, correspondingly.

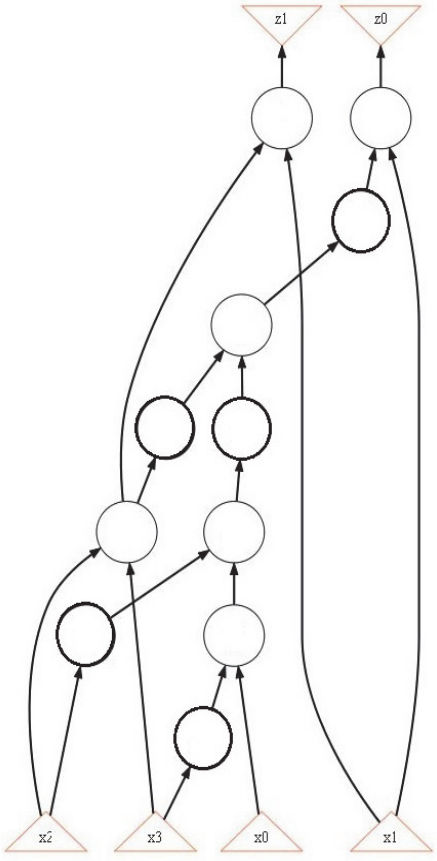

Figure 1: A circuit *S*.

As an example, consider a combinational circuit in Fig. 1 with a set $X = \{x_0, x_1, x_2, x_3\}$ of inputs, a set $Z = \{z_0, z_1\}$ of outputs and 11 AND and NOT gates (AIG nodes); the latter are taken in bold.

Definition 2. By definition, a logic circuit *implements* or *represents* a system of Boolean functions. A circuit accepts a Boolean vector as an input and produces a Boolean vector as an output according to the corresponding system of Boolean functions. Each logic circuit can be described by a Look-up-Table (LUT). A LUT contains a set of input/output pairs of a given circuit: if for the input **i** the circuit produces an output **o**, then the pair **i**/**o** is included into the LUT.

A LUT can be used as the specification when deriving a logic network that implements the system of Boolean functions, and there exist a number of methods how to synthesize a logic network that implements a given system of functions. In this paper, we use the ABC tool (Berkeley Logic Synthesis and Verification Group, ABC) to design a circuit for a given LUT. For this purpose, such LUT is described in a special form; in this paper, we use the PLA format.

As in this paper we focus on using logic networks to evaluate/predict the QoE of a given service, we further briefly sketch the algorithms proposed in (Kushik et al., 2014) for deriving and training these circuits. In order to derive the initial circuit *C*, one uses statistical data gathered from service experts, from the automatic evaluation of service parameters and/or from end-users, who have experience of using the service. These statistical data are encoded as Boolean vectors of appropriate length, and this set of input/output vectors is written in the PLA format. The circuit *C* that evaluates the QoE value is then designed from a system of partially specified Boolean functions. The corresponding procedure is given as Algorithm 1.

Algorithm 1 for deriving an initial logic circuit to evaluate the QoE value

Inputs: Service parameters $p_1, p_2, \ldots, p_k$ with nonnegative (unsigned) integer values bounded by $M_{p_1}, M_{p_2}, \ldots, M_{p_k}$; maximal value of the QoE M_{QoE};

Statistical data, i.e., feedbacks from users $U_1, \ldots, U_r$ represented as a list of patterns $p_1\_value, p_2\_value, \ldots, p_k\_value, UserSatisfaction\_value$.

Output: a logic circuit *C*

1. Determine the number of primary inputs and primary outputs of *C*:

The number of primary inputs equals $\sum_{i=1}^{k}]\log_2 M_{p_i}[$ while the number of primary outputs equals $]\log_2 M_{QoE}[$.

2. Derive a LUT

2.1 For each user U_i, $i \in \{1, \ldots, r\}$, convert his/her statistic scores $p_1\_value, p_2\_value, \ldots, p_k\_value, UserSatisfaction\_value$ into Boolean vectors and add the corresponding lines to the LUT.

3. Synthesize the circuit *C* from a system of partial Boolean functions and **Return *C***.

The circuit C has to be self-adaptive, i.e., when a new end-user agrees to leave his/her feedback about the service quality the circuit behavior has to be modeled under a corresponding input **i** and if the result produced by the circuit differs significantly from the expected then the circuit has to be resynthesized. To evaluate the difference between the circuit output and the user satisfaction value Kushik et al. introduced some value τ that represents a confidence interval, i.e., the $QoE(W)$ produced by the circuit C has to belong to the interval [*UserSatisfaction_value* – τ, *UserSatisfaction_value* + τ]. If this fact does not hold, i.e., $|QoE(W) - UserSatisfaction\_value| > \tau$ then the circuit C is resynthesized. The corresponding procedure taken from (Kushik et al., 2014), is presented as Algorithm 2.

Algorithm 2 for learning / training the logic circuit that evaluates / 'predicts' the QoE value for a service
Inputs: QoE parameters p_1, p_2, …, p_k with nonnegative values bounded by M_{p_1}, M_{p_2}, …, M_{p_k}; maximal value of the QoE M_{QoE};
The circuit C that evaluates the QoE value for a service W;
A new user feedback p_1*_value*, p_2*_value*, …, p_k*_value*, *UserSatisfaction_value*;
Maximal difference τ for corresponding confidence interval.
Output: a modified logic circuit C
1. Integers p_1*_value*, p_2*_value*, …, p_k*_value*, *UserSatisfaction_value* are converted into Boolean vectors **v_p1, v_p2, …, v_pk, v_us**.
2. The output $QoE(W)$ of the circuit C is computed for the input **v_p1, v_p2, …, v_pk**.
3. If $|QoE(W) - UserSatisfaction\_value| > \tau$ then
3.1 *If* the line **v_p1, v_p2, …, v_pk** is specified as input in the LUT, then change the corresponding output into **v_us,**
Otherwise
Add the new line **v_p1, v_p2, …, v_pk, v_us** to the LUT.
3.2 Synthesize the new circuit C'; assign $C = C'$ and **Return** C.

In this paper, we propose an approach how a circuit that predicts the QoE of a composite service can be derived under the assumption that the QoE of the service components are given. These circuits can be derived using Algorithm 1 and effectively trained by applying Algorithm 2. The approach proposed in the paper is illustrated by a running example.

3 A RUNNING EXAMPLE

In this paper, we consider a given web service as a running example. In particular, we rely on the example of *vacation planner* service that is taken from (Kondratyeva et al., 2013). This service offers a user an opportunity to purchase flight tickets and to book an accommodation at the destination point. A user submits traveling dates and the planner proposes a number of available options for flight tickets and hotel rooms. If the user and the planner agree on the flight ticket and the hotel room then the vacation is successfully booked. Otherwise, the vacation is not reserved. The list of crucial service parameters that significantly affect the QoE is as follows: the execution time, service availability and service popularity. In other words, the QoE of the vacation planner significantly depends on the component values of the vector $\langle t, a, p \rangle$, where t denotes the execution time, a – the availability and p – the popularity.

As the vacation planner is designed as a composition of a flight booking and a hotel booking services, the QoE of this composite service can be calculated based on the QoE of the flight booking and the QoE of the hotel booking services. Given the flight booking service, in the running example, we consider that the execution time t and its popularity p are the crucial parameters for most users. Let Table 1 contain the statistical data gathered from the users and/or experts A, B, C, and D.

Table 1: Statistical data gathered for the flight booking service.

User identifier	t	p	QoE
A	3	0.3	3
B	1	0.9	5
C	3	0.2	1
D	2	0.5	4

Similar to the flight booking service, in this paper, we consider the availability a to be a crucial parameter for the second component of the vacation planner. In other words, once a user has agreed on all the flights details, he/she is redirected to a hotel booking service that has to be necessarily available at the moment. If this service is not available the user's QoE goes immediately down. The corresponding statistical data left by experts and/or some users E and F of the hotel booking service are shown in Table 2.

Given the statistical data for the service components, we consider that the QoE of the composite service is always the minimal value for

Table 2: Statistical data gathered for the hotel booking service.

User identifier	*a*	QoE
E	0.9	5
F	0.6	4

all possible values of the vector $\langle t, a, p \rangle$. The latter means, that in order to predict the QoE of the vacation planner, one should consider the worst users' opinions. The reason is that if a user is not satisfied with a given service component, he/her satisfaction cannot be increased with the use of other components. In the running example, in order to consider the statistical data for the vacation planner one should concatenate the data given in Tables 1 and 2, correspondingly. The resulting statistical data are given in Table 3.

Table 3: Statistical data for the vacation planner.

t	*p*	*a*	QoE
3	0.3	0.9	3
1	0.9	0.9	5
3	0.2	0.9	1
2	0.5	0.9	4
3	0.3	0.6	3
1	0.9	0.6	4
3	0.2	0.6	1
2	0.5	0.6	4

Table 3 contains eight lines; each line represents a vector $\langle t, a, p, \text{QoE} \rangle$ where the QoE is the minimal value taken from the vectors $\langle t, a, \text{QoE} \rangle$ (Table 1) and $\langle p, \text{QoE} \rangle$ (Table 2).

Consider two logic circuits C_1 and C_2 designed for predicting the QoE of the flight booking and the hotel booking services, correspondingly. We further discuss how one can build a logic circuit that predicts the QoE value of the vacation planner.

4 SCALABLE APPROACH FOR ESTIMATING THE QoE OF A COMPOSITE SERVICE

In this section, an approach for automatic evaluation/'prediction' of the QoE value for a composite service is proposed. Without loss of generality, we consider two service components S_1 and S_2 that are somehow combined when designing the composite service $S_1 @ S_2$, where @ is a composition operator. If the number k of service components is greater than two, this approach can be applied iteratively, i.e. first, the QoE of the service $S_1 @ S_2$ is estimated, then, the QoE of the service $(S_1 @ S_2) @ S_3$ is estimated, etc. At the final step, the QoE is predicted for the service $(S_1 @ \ldots @ S_{k-1}) @ S_k$. The question about communicative and associative properties of the composition operator is out of the scope of this paper.

Given two composite services S_1 and S_2, consider two logic circuits C_1 and C_2 that predict their QoE values, correspondingly. These circuits can be derived as proposed in (Kushik et al., 2014). We provide an algorithm for designing a logic circuit $C_1 @ C_2$ that predicts the QoE value of the composition $S_1 @ S_2$.

4.1 Deriving a Logic Circuit for Predicting the QoE of a Composite Service

In this section, we provide an algorithm (Algorithm 3) for designing a logic circuit $C_1 @ C_2$. At the first step, the set of inputs of this circuit is determined. In fact, this set contains all the inputs that correspond to S_1 service parameters and S_2 service parameters. In other words, the set of inputs for $C_1 @ C_2$ is the union of the sets of inputs for C_1 and C_2, If the sets of S_1 and S_2 parameters do not intersect, the set of inputs for $C_1 @ C_2$ is the set of inputs for C_1 plus inputs of C_2.

At the second step, the special circuit C_{min} for implementing a *minimum* function is designed. This circuit will be used to choose between two QoE values produced by the circuits C_1 and C_2. As mentioned above, we always rely on the minimal value of the two QoE values, considering that the user satisfaction can be only decreased for a composite service. Each circuit C_1 or C_2 produces the Boolean vector of corresponding length. These vectors correspond to integers $\mathbf{I_1}$ and $\mathbf{I_2}$ that represent the QoE values for the service components S_1 and S_2. The MIN function is used to choose the minimum value of $\mathbf{I_1}$ and $\mathbf{I_2}$; if these values coincide then the QoE of the composite service equals $\mathbf{I_1} = \mathbf{I_2}$. The corresponding circuit that implements this function has the number of inputs that is the sum of outputs of circuits C_1 and C_2. Hereafter, in the paper, we consider that the QoE is measured within the Mean Opinion Score (MOS) scale (ITU-T, 2006) and thus, outputs of each circuit encode integers of the set {1, 2, 3, 4, 5}, i.e., the number of outputs of each circuit C_1 and C_2 equals three.

At the final step of the algorithm, the outputs of the circuits C_1 and C_2 are connected to the inputs of the circuit C_{min}, and the resulting circuit is returned.

A scheme that illustrates the procedure for

deriving the circuit C_1 @ C_2 for evaluating the QoE of the composed service is shown in Fig. 2. The items of the set P correspond to Boolean vectors which represent the values of parameters p_1, p_2, …, p_k of the service S_1 whereas the items of the set Q correspond to Boolean vectors which represent the values of parameters q_1, q_2, …, q_l of the service S_2. The set $P \cap Q$ corresponds to the Boolean vectors, which represent the same parameters of services S_1 and S_2. Therefore, the set $P' = P \setminus P \cap Q$ denotes the set of Boolean vectors for parameters of S_1 that are not shared with S_2 while the set $Q' = Q \setminus P \cap Q$ denotes the set of Boolean vectors for parameters of S_2 that are not shared with S_1.

Algorithm 3 for deriving a circuit C_1 @ C_2

Inputs: Service components S_1 and S_2.

S_1 has the set $P = \{p_1, p_2, \ldots, p_k\}$ of parameters; each p_i parameter value is bounded an integer M_{p_i}.

S_2 has the set $Q = \{q_1, q_2, \ldots, q_l\}$ of parameters; each q_i parameter value is bounded an integer M_{q_i}.

The circuit C_1 has $\sum_{i=1}^{k}]\log_2 M_{p_i}[$ inputs and three outputs; the circuit C_2 has $\sum_{i=1}^{l}]\log_2 M_{q_i}[$ inputs and three outputs.

Output: a logic circuit C_1 @ C_2.

1. Determine the number of primary inputs of C_1 @ C_2:

The number of primary inputs equals $(\sum_{i=1}^{k}]\log_2 M_{p_i}[+ \sum_{i=1}^{l}]\log_2 M_{q_i}[) - \sum_{i=1}^{t}]\log_2 M_{g_i}[$ for all g_i that belong to the $P \cap Q$, where $|P \cap Q| = t$. The number of primary outputs of the circuit of C_1 @ C_2 equals three.

2. Design the circuit C_{min}. This circuit has six inputs i_1, i_2, ... i_6, and returns the minimal value of two integers $\mathbf{I}(i_1 i_2 i_3)$ and $\mathbf{I}(i_4 i_5 i_6)$.

3. Synthesize the circuit $C = C_1$ @ C_2 identifying inputs which correspond to the same parameters of services S_1 and S_2; the outputs of C_1 are connected to inputs i_1, i_2, i_3 of C_{min} while the outputs of C_2 being connected to the inputs i_4, i_5, i_6 of C_{min},

Return C.

The circuit C_{min} in Fig. 2 is used to compute the minimal value of the two QoE values computed by the circuits C_1 and C_2 for the services S_1 and S_2, correspondingly. The set I of C_{min} denotes the set of Boolean vectors representing the QoE of the composite service of S_1 and S_2.

By construction of the circuit C_1 @ C_2 using Algorithm 3, the following proposition holds.

Proposition 1. Given a composite service S_1 @ S_2 and two statistical patterns p_1_value, p_2_value, …, p_k_value, $S_1\_UserSatisfaction\_value$, and q_1_value, p_2_value, …, q_l_value, $S_2\_UserSatisfaction\_value$. Algorithm 3 produces the output $C = C_1$ @ C_2 such that the output **o** of the circuit C corresponds to the minimum of the integers $S_1\_UserSatisfaction\_value$ and $S_2\_UserSatisfaction\_value$.

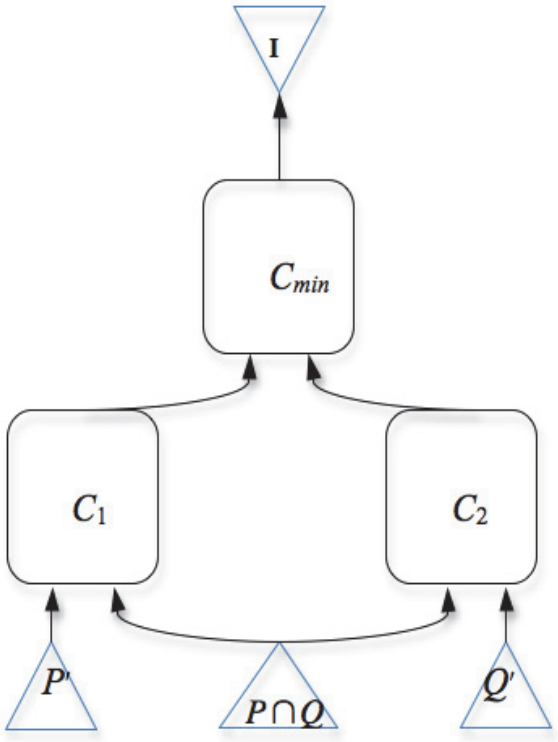

Figure 2: A scheme to derive the circuit C_1 @ C_2, where $P' = P \setminus P \cap Q$ and. $Q' = Q \setminus P \cap Q$.

We notice that the complexity of Algorithm 3 is polynomial as it is mostly 'hidden' in Step 3. The arithmetic evaluation of the number of primary inputs and outputs (Step 1) of the circuit C_1 @ C_2 can be performed in 'no time' while the circuit C_{min} can be derived just once for various service components S_1 and S_2. Therefore, the complexity of Algorithm 3 can be estimated as the number of operations required to connect each output of circuit

C_1 (or C_2) to a corresponding input of the circuit C_{min}, and these operations are very scalable. The latter proves the scalability of the overall approach.

As mentioned above, the proposed approach to estimate the QoE of a composite service can be also applied when there exist more than two component services. For example, when evaluating the QoE of the service S that is represented as composition (S_1 @ S_2) @ S_3 of three services, one can apply the proposed approach iteratively. At the first step, the QoE of the composition S_1 @ S_2 is predicted by the circuit C_1 @ C_2. At the second step, the circuit C = (C_1 @ C_2) is combined with the circuit C_3 using again Algorithm 3. Let the set R correspond to Boolean vectors which represent the values of parameters r_1, r_2, …, r_m of the service S_3. In this case, the set of inputs of the circuit (C_1 @ C_2) @ C_3 is the union of the sets P, Q, and R of the circuit components. After the first application of Algorithm 3, the union W of the sets P and Q is obtained, i.e., $W = P \cup Q$. After the second Algorithm 3 application, the circuit C = (C_1 @ C_2) @ C_3 is obtained, and the set of its inputs is $W \cup R$. A scheme that illustrates the procedure for deriving the circuit (C_1 @ C_2) @ C_3 when evaluating the QoE of the composed service is shown in Fig. 3.

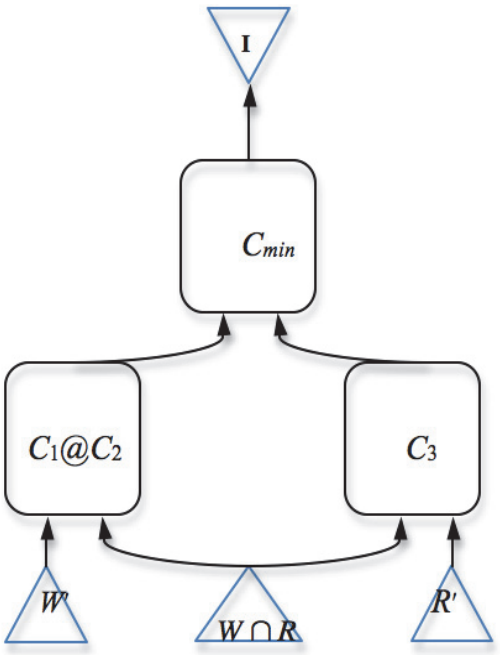

Figure 3: A scheme to derive the circuit (C_1 @ C_2) @ C_3, where $W' = W \setminus W \cap R$ and. $R' = R \setminus W \cap R$.

4.2 Designing a Logic Circuit C_{min} by ABC

The complexity of the circuit C = C_1 @ C_2 significantly depends on the complexity of the circuit C_{min}. We have derived this logic network using the software tool ABC (Berkeley Logic Synthesis and Verification Group, ABC). For this purpose, we have derived a LUT for a corresponding MIN function. This LUT contains 64 lines, as the circuit has 6 inputs. The corresponding LUT is partially presented in Table 4. The circuit C_{min} has *significant* input values that correspond to pairs (j, k) of integers, j, $k \in \{1, 2, 3, 4, 5\}$. Other pairs with integers 0, 6, 7 are so-called Don't Care (DNC) inputs, and as the circuit is used to compute the minimum of two integers, for these pairs, we define the output as the corresponding minimal value, extending the input domain of the corresponding MIN function.

We have run the ABC tool against the LUT that

Table 4: A LUT for the circuit Cmin.

$x_1 x_2 x_3 x_4 x_5 x_6$	MIN
000 000	000
000 001	000
000 010	000
000 011	000
…	…
001 110	001
001 111	001
010 000	000
010 001	001
010 010	010
010 011	010
010 100	010
010 101	010
010 110	010
…	…
100 000	000
100 001	001
100 010	010
100 011	011
100 100	100
100 101	100
100 110	100
…	…
110 100	100
110 101	101
110 110	110
…	…
111 100	100
111 101	101
111 110	110
111 111	111

is partially represented in Table 4. For this purpose, we have presented the set on input/output vectors in the PLA format. The resulting circuit C_{min} designed by the ABC has 40 AIG nodes (gates).

We mention that the size of the circuit C_{min} is essentially related to the scalability of the proposed approach and the circuit C_{min} came out to be very compact and thus, can be effectively combined with the circuits C_1 and C_2. Moreover, the size of C_{min} is very close to the size of the circuits that can be obtained when predicting the quality of some 'real life' services. As an example, the reader can address the experimental results for multimedia services presented in (Kushik et al., 2014), where the size of the circuit with two service parameters, namely jitter and packet loss, was 154 AIG nodes.

Nevertheless, as various services are designed for different purposes and, therefore, have different crucial parameters, we note that further experimental research is needed to estimate the efficiency of the proposed approach.

6 DISCUSSION ON APPLICABILITY OF THE APPROACH

In this section, we briefly discuss how the proposed approach for the composite service QoE evaluation can be more rigorously implemented. In the previous sections, we considered the worst case scenario when the QoE value is the minimal value of QoE over all component services. However, this assumption is very strict and not realistic in many cases. More often, the QoE of the composite service significantly depends on the structure of the composite service and can be estimated as a special formula taken into consideration the service composition pattern. As usual, a linear combination of the two variables QoE_1 and QoE_2 (or more if there are more component services) that represent the QoE values of the services C_1 and C_2 can be considered as the simplest case. In this case, following the technique proposed in the paper, one should derive a logic circuit $C_{formula}$ that substitutes the C_{min} one and implements a corresponding linear combination. Consider a circuit $C_{formula}$ that returns the Boolean vector $\mathbf{o} = (o_1o_2o_3)$ that corresponds to the integer that is calculated with a formula (α_1 $\mathbf{I}(i_1i_2i_3)$ + α_2 $\mathbf{I}(i_4i_5i_6)$). The coefficients α_1 and α_2 can been taken from various domains, however, in order to simplify the logic synthesis procedure they should be normalized as integer values. A modified scheme that illustrates the procedure for deriving the circuit C_1 @ C_2 such that the QoE of the overall circuit is computed as the linear combination (α_1 $\mathbf{I}(i_1i_2i_3)$ + α_2 $\mathbf{I}(i_4i_5i_6)$), is shown in Fig. 4.

The circuit $C_{formula}$ that computes the linear combination (α_1 $\mathbf{I}(i_1i_2i_3)$ + α_2 $\mathbf{I}(i_4i_5i_6)$) in the circuit C_1 @ C_2, can be implemented in different ways. Nevertheless, this implementation is reduced to implementing two arithmetical multiplications and one addition.

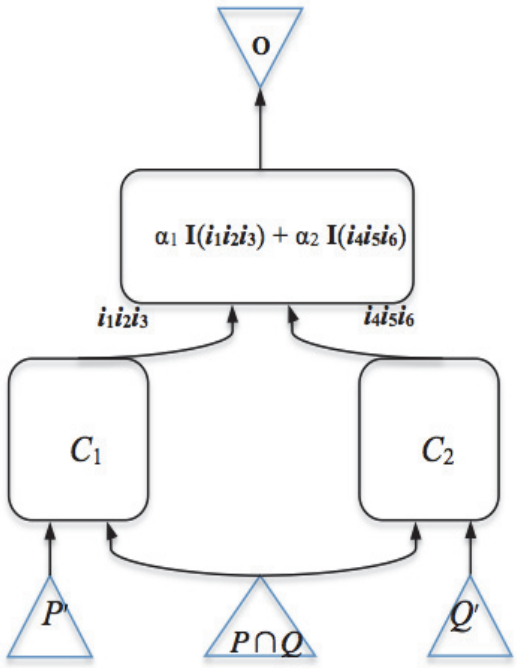

Figure 4: A modified scheme to derive the circuit C_1 @ C_2, where $P' = P \setminus P \cap Q$ and $Q' = Q \setminus P \cap Q$.

In this case, the most scalable implementation can be achieved when the coefficients α_1 and α_2 are integers that represent the powers of two, namely, there exist $x > 0$ and $y > 0$, such that $\alpha_1 = 2^x$ and $\alpha_2 = 2^y$. This fact simplifies the multiplication procedure. Indeed, the circuit that performs such multiplication can be implemented as a shift register that shifts the inputs $i_1i_2i_3$ and $i_4i_5i_6$ by x and y bits, correspondingly. Therefore, such linear combinations preserve the scalability of the proposed approach. However, the use of different coefficients can reduce the approach scalability. This drawback can be overcome by considering α_1 and α_2 as external inputs of the $C_{formula}$. Similar to Section 4, the circuits can be constructed not for two but for bigger number of service components. More general types of the circuit $C_{formula}$ that implement some specific

functions that compute the QoE value of the composite service and take into account the compositional pattern as well as the component QoE values need additional research and are left as future work.

6 CONCLUSIONS

In this paper, we have proposed an approach for scalable QoE prediction of a composite service. The approach relies on logic circuits that are designed to predict the QoE values of the service components. The algorithm provided in the paper returns the logic circuit that predicts the QoE value of a composite service taking into account the fact that the user satisfaction can be only decreased in the service composition. Therefore, a MIN function can be effectively used to decide between the two QoE values of the service components. We have estimated the complexity of the resulting circuit that predicts the QoE of the composite service. Preliminary experimental results show the scalability of the proposed approach. More experiments with different services considering different service parameters are planned as a future work.

We also notice that despite the fact that using the worst-case scenario provides a scalable approach for the QoE composition estimation, in many realistic cases, the internal composition structure, i.e., compositional patterns have to be taken into account. The reason is that the degradation of the QoE in one component can affect the QoE of other components in different ways. On the other hand, a user satisfaction within a composite service cannot rely only of the values of the service component parameters, it also depends on the network traffic, the properties of the computer of the user, additional user parameters such as his/her mood, etc. The approach proposed in the paper does not take into account the above issues, and this study is also remained for the future work.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the scientific support of the research group lead by Prof. Ana Cavalli (TELECOM SudParis, France) that initiated the study of the QoE estimation and was significantly involved in the first steps of using the logic synthesis techniques for the service analysis issues. The authors are pleased to provide novel contributions to this area based on these first steps that have been made together.

The authors also mention that this work is partially supported by RFBR grant № 14-08-31640 мол_а (Russia).

REFERENCES

Ahmed, S., Begum, M., Hasan Siddiqui, F., Abul Kashem, M., 2012. Dynamic Web Service Discovery Model Basedon Artificial Neural Network with QoS Support. *International Journal of Scientific & Engineering Research Volume 3, Issue 3*, *pp. 1-7*.

Al-Masri, E., Mahmoud Qusay, H., 2009. Discovering the Best Web Service: A Neural Network-based Solution. *SMC 2009, pp. 4250-4255*.

Berkeley Logic Synthesis and Verification Group, ABC: A System for Sequential Synthesis and Verification, *url: http://www.eecs.berkeley.edu/~alanmi/abc/*.

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D., 2004. Web services architecture. *W3C Working Group Note, W3C Technical Reports and Publications, url: http://www.w3.org/TR/ws-arch/*.

El Hadad, J., Manouvrier, M., Rukoz, M., 2010. TQoS: Transactional and QoS-Aware Selection Algorithm for Automatic Web Service Composition. *IEEE Transactions on Services Computing, vol. 3, issue. 1, pp. 73-85*.

Kondratyeva, O., Kushik, N., Cavalli, A., Yevtushenko N., 2013. Evaluating Web Service Quality using Finite State Models. In *Proc. of QSIC 2013*.

Kushik, N., Pokhrel J., Yevtushenko N., Cavalli A.R., Mallouli W., 2014. QoE Prediction for Multimedia Services: Comparing Fuzzy and Logic Network Approaches. *International Journal of Organizational and Collective Intelligence, 4(3), pp. 44-65*.

Lin, M., Xie, J., Guo, H., Wang, H., 2005. Solving QoS-driven web service dynamic composition as fuzzy constraint satisfaction. *EEE 2005, pp. 9-14.*

Mitchell, T.M., 1997. *Machine learning*. McGraw Hill series in computer science, McGraw-Hill.

Pokhrel, J., Mallouli, W., Montes de Oca, E., 2013. QoE Prediction and Self-Learning Mechanisms. *Technical report on the PIMI Project*.

Pokhrel, J., Wehbi, B., Morais, A., Cavalli, A., Allilaire, E., 2013. Estimation of QoE of video traffic using a fuzzy expert system. In *Proc. of CCNC, pp. 224-229*.

Torres, R., Astudillo, H., Salas, R., 2011. Self-Adaptive Fuzzy QoS-Driven Web Service Discovery. In *IEEE SCC 2011, pp. 64-71*.

ITU-T, 2006. Mean opinion Score (MOS) terminology. Recommendation P.800.1.

Winckler, M.A., Bach, C., Bernhaupt, R., 2013. Identifying user experience dimensions for mobile incident reporting in urban contexts. *IEEE Transactions on Communications, vol. 56, no. 2, pp. 40-82*.

Zadeh, L.A., 1965. Fuzzy sets. *Information and Control, 8 (3), pp. 338–353*.

Zheng, H., Zhao, W., Yang, J., Bouguettaya, A., 2013. QoS analysis for web service compositions with complex structures. *IEEE Transactions on Services Computing, vol. 6, issue. 3, pp. 373 - 386*.

Towards an Opportunistic, Socially-driven, Self-organizing, Cloud Networking Architecture with NovaGenesis

Antonio M. Alberti[1], Waldir Moreira[2], Rodrigo da Rosa Righi[3], Francisco J. Pereira Neto[1], Ciprian Dobre[4] and Dhananjay Singh[5]

[1]*ICT Laboratory, Instituto Nacional de Telecomunicações - INATEL, Santa Rita do Sapucaí, Minas Gerais, Brazil*

[2]*COPELABS, Univ. Lusófona, Lisbon, Portugal*

[3]*PIPCA, Universidade do Vale do Rio dos Sinos, São Leopoldo, Rio Grande do Sul, Brazil*

[4]*Computer Science Dpt., Univ. Politechnica of Bucharest, Bucharest, Romania*

[5]*Electronics Eng. Dpt., Hankuk Univ. of Foreign Studies,Yongin, South Korea*

alberti@inatel.br, waldir.junior@ulusofona.pt, rrrighi@unisinos.br, ciprian.dobre@cs.pub.ro, dsingh@hufs.ac.kr

Keywords: Future Internet, Opportunistic, Cloud Networking, Socially-driven, Self-organizing, Software-as-a-Service.

Abstract: The exponential growth on the number of mobile devices and their capabilities are leveraging new possibilities of networking architectures for processing, storing, and exchanging of information. At a glance, existing architectures take advantage of these devices, the social behavior of their users, and/or the dynamicity on resource usage. Despite of the potential of existing initiatives, they do not interoperate which reduce their applications and deployment. As we walk towards a very dynamic world (regarding the user needs and characteristics, the information traversing the network, and the networking capability to adaptation at both users features and content of the demands levels), these architectures should merge into a solution that fits any type of scenario. In this paper, we specify an opportunistic, socially-driven, self-organizing, cloud networking architecture using a future Internet proposal named NovaGenesis. We highlight the requirements and solutions that NovaGenesis brings to accommodate the inherent challenges of today's dynamic networking scenario. Thus, we describe a convergent architecture, which integrates the new requirements with the already implemented NovaGenesis features.

1 INTRODUCTION

The way that users communicate nowadays is very different from a few years ago. The classic host-based paradigm, in which users would request specific content stored in specific locations, is giving room to new forms for users to exchange data. This is a product of not only devices becoming very portable, but also gathering the latest advancements in terms of processing, storage, and wireless technologies. Thus, users are able to produce and consume content anywhere and anytime, and such data exchange may take place through spontaneously formed networks. Furthermore, this content can be stored locally and/or on the users' personal clouds, as well as on public clouds.

Different networking approaches have emerged. Delay/disruption-tolerant networking (DTN) (Caini et al., 2008) deals with scenarios where intermittent connectivity is rather common among network nodes. Opportunistic Networking (ON) (Moreira et al., 2012) exploits different contact opportunities among users to exchange data. Cognitive Radio Network (CRN) (Ahmed et al., 2010), aims at exploring radio frequency spectrum holes to accommodate communication links. Information-Centric networking (ICN) (Xylomenos et al., 2014) emerged from the idea that people care for content itself no matter where it is, decoupling content identification ("what") from its location. User-centric networking (UCN) (Sofia and Mendes, 2008) includes user-provided devices and systems to build social-driven networks. These approaches are being used to address different challenges (e.g., intermittent connectivity, high mobility, longer delays, expensive infrastructure and/or connectivity) of emerging access networks. Also, cloud and big data systems have been integrated to these networking approaches as they can further increase the capabilities of the user devices and handle the vast amount of data that users produce.

One can observe that these networking approaches comprise a wide variety of data exchange and processing approaches that, despite of the already known

potential, still operate for specific purposes and do not interoperate as required. Users are not so interested in the technicalities and employed approaches that allow them to exchange information. Instead, users expect an integrated cloud/networking infrastructure that allows them to share information directly with other peers, without relying on infrastructure and/or expensive connectivity services.

Despite the fact that existing networking and information access approaches share few concepts and concerns, they fail to cohesively integrate the aforementioned technologies (Alberti, 2012). We believe that such approaches can neatly come together as to form a converged architecture. Our starting point is NovaGenesis (www.inatel.br/novagenesis) initiative, which is a clean slate convergent information architecture (CIA) being developed by our team. By CIA, we mean an architecture that integrates information exchanging with processing and storage. It can be seen as a generic architecture, where Internet is converged to cloud computing and big data. NovaGenesis already integrates ICN, CRN, service-oriented architecture (SOA) (Papazoglou et al., 2007), software-defined networking (SDN) (McKeown et al., 2008), and Internet of things (IoT) (Conti, 2006). NovaGenesis paves the way to a CIA that advances integration efforts to include ON, ICN, UCN, and cloud.

In this context, the main aim of this work is to fulfil the need for a solution of converged architectures that allows for the application of different networking paradigms in a neat way. Each of these key ingredients strongly contributes to advance a specific designing dimension. When two or more of these ingredients are synergistically integrated, there is a "cross fertilization", a catalyzing effect, which favor global architectural advances instead of local ones. NovaGenesis is explored as the foundation for this architecture. New services are proposed to be integrated to NovaGenesis proposal. We contribute with a novel approach towards user-centric architectures.

This paper is structured as follows: we start by briefly overviewing the relevant networking paradigms to be considered by our convergent architecture in Section 2. In Section 3, we showcase NovaGenesis software-based CIA, the architecture we chose as the foundation of our social-driven initiative. Section 4 presents our contribution, showing a qualitative analysis regarding how NovaGenesis addresses the challenges behind the proposed architecture, pointing perspectives, pre-requirements, and open issues. Finally, Section 5 concludes the paper, also highlighting some future work.

2 RELEVANT NETWORKING PARADIGMS

This section presents the different paradigms that shall be comprised by our convergent architecture.

Opportunistic Networking (ON) exploits the contact opportunities taking place among users' devices to allow data exchange. Opportunistic forwarding can be seen from two perspectives regarding user social behavior, namely social-oblivious and social-aware approaches. From these approaches, the latter has gained much attention of the research community given the fact that social information is less volatile (i.e., changes less, favoring data exchange) than mobility (Moreira and Mendes, 2013). Our convergent architecture shall take into consideration the dynamism of user social behavior found in their daily routine in order to properly infer the different levels of social interactions among users and the interests of these users (Moreira et al., 2012) (Ciobanu et al., 2013) (Ciobanu et al., 2014b) (Moreira et al., 2014) since the dynamics of user social behavior does have an impact on the performance of opportunistic forwarding (Moreira and Mendes, 2015a) (Moreira and Mendes, 2015b). With that, our architecture is expected to provide the users with data exchange opportunities over only socially relevant links, thus with improved delivery probability while reducing associated cost and experienced latency.

With **Content-Centric Networking (CCN)**, data traverses the network according to the match between its name and the interests that users may have in such contents, independently of its location, resulting in an efficient, scalable, and robust content delivery. There are different efforts for defining a CCN architecture (DONA, NDN, NetInf), each with its own particularities (e.g., employ their own naming scheme) and looking into different CCN aspects (e.g., naming, security, routing, caching, transport) according to the application to which they have been devised (Xylomenos et al., 2014). With the advances in technology, devices have become more portable and with increased capabilities (e.g., processing, storage). In such dynamic networking scenarios, users are prosumers (i.e., producers and consumers) of information with a high demand to share/retrieve content anytime and anywhere, independently of the intermittency level of wireless connectivity, dynamic behavior of users, physical obstacles, lack of cooperation, closed (i.e, secured) networks, among others. Given its potential, the convergent architecture shall incorporate CCN features (end-to-end path abstraction, interest-based content-driven dissemination) as to cope with the dynamicity of today's networking

scenario. Thus, our architecture shall provide the content that users demand based on their interests which they propagate while carrying on their daily routines.

User-Centric Networking (UCN) focuses on the user who is the main pillar for routing, security, and among other networking aspects (Sofia et al., 2014). This networking paradigm empowers the user that can easily provide services (e.g., connectivity, printing) to others. Within this context, the user besides producing and consuming content as mentioned before, now becomes a micro-provider (Sofia and Mendes, 2008) changing currently known Internet communication models (end-users comprise a user-provided network extending services, where user willingness in sharing resources/services allows scalable services).

There are different approaches that relate to user-provided networking spanning a vast range of applications: making use of proprietary equipment to share connectivity (SparkNet at htpp://sparknet.fi/); simply turning the end-user device into a sharing point (Whisher/WiFi.Com at http://www.whisher.com); creating a network for sharing resources (Wray Village' wireless broadband at http://www.infolab21.co.uk/livinglab). However, it is important to note that these approaches aim solely at sharing connectivity. This is indeed a type of resource that users are very much interested, but there is more to it. The ULOOP (siti.ulusofona.pt/˜uloop/) project clearly highlights this: ULOOP users are provided with th means to exchange resources as they wish based on the trust levels between these users and/or based on the exchange of virtual currency. The project considers they dynamic behavior of users to allow the exchange of different types of resource beyond connectivity. With this in mind, our convergent architecture aims at providing users with the means of sharing the resources they have the most and make use of resources they require at a given moment. By combining opportunism with content centricity, the convergent architecture is expected to further empower the users allowing them to naturally engage in the system y providing and consuming resources according to their current demands.

Cloud Computing Elasticity exploits the fact that resource allocation is a procedure that can be performed dynamically according to the demand for either the service or the user (Jamshidi et al., 2014). Our convergent architecture is expected to increase the number of network resources (e.g., routing elements, pre-processor nodes and gateways) in order to provide and keep a service level agreement (SLA) between the user and the Internet architecture assembled above the cloud. Virtual machine migration, addition and resizing are techniques that could be combined to offer an elasticity semantic for this novel Internet architecture. Additionally, we envision the reduction of subnet resources when the network demand is moderated, so contributing to implement green computing with energy saving (i.e., consolidation technique, shutting down VMs and the host node)

3 NovaGenesis (NG) ARCHITECTURE

NovaGenesis (Alberti et al., 2014) project started in 2008 to address this question: imagine there is no Internet architecture right now, how could we design it using the best contemporary technologies? We selected several technologies to best implement NovaGenesis design principles, looking for deep synergies among them. NovaGenesis can be defined as a convergent information architecture (CIA). By CIA we mean an architecture that synergistically integrates information processing and storage (as contended by cloud computing), as well as information exchanging (like the current Internet architecture or other emerging networks, e.g., software defined networks or mobile terminal networks). All the concepts presented in this section are summarized in Figure 1.

3.1 Names, Identifiers, and Locators

The NG cornerstone is naming. A *name* is a set of symbols that denote something, some existence. It is deeply rooted in language. For example, one can use the name "Paris" to denote a city in Europe. The same name can be used to denote many different existences, e.g. "Paris" is also the name of a famous north American socialite. In this context, an important decision choice we did was: what existences need to be named on a CIA? By existence we mean everything that simply is. People love to name everything - from cars to airplanes, applications to computers, photos to movies, etc. Additionally, an important requirement for future architectures is that they should be able to better "understand" the meaning of the language used by people - which is called semantic technology.

Many notable companies are investing on semantic computing. Examples are IBM's Watson and Google's Brain. With the advent of the Internet of things (IoT) - where virtually anything can belong to the Internet - we assumed that all possible entities could be named, bringing machines closer to people natural language. Naming should be very flexible and broad. However, not all natural language names (NLNs) are adequate for efficient and safe naming. Therefore, we adopted a second kind of naming in

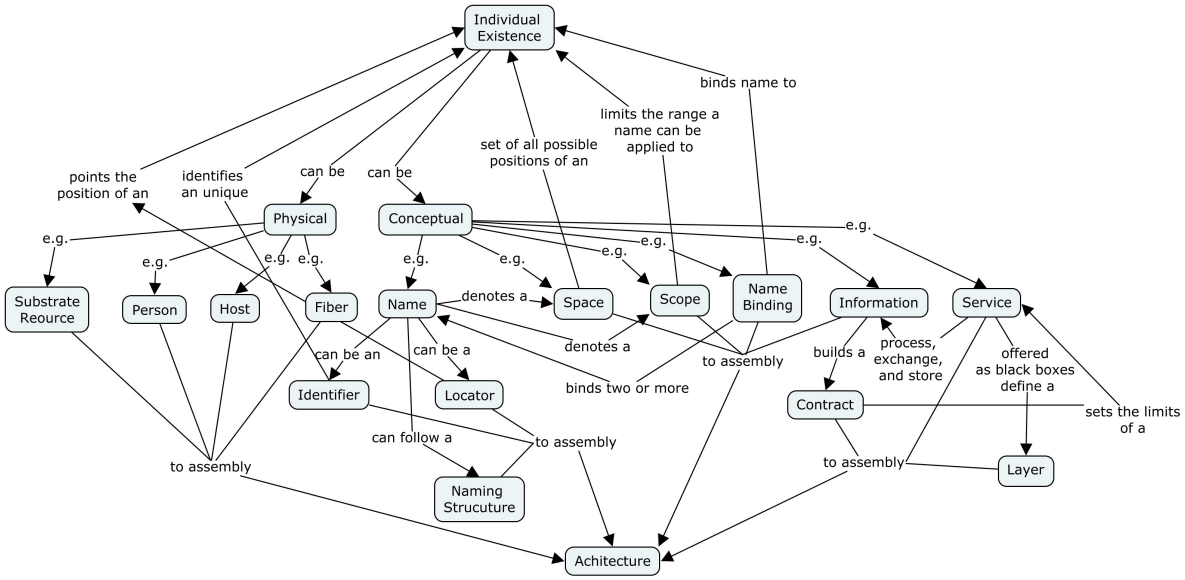

Figure 1: NovaGenesis concepts for convergent information architecture.

NG architecture. The so called *self-certifying names* (SCNs). A SCN is typically obtained by passing a input binary pattern by a hash function. The pattern can be the entity itself, e.g., a chunk of data of a photo, or a digital representation of some physical world attributed, e.g., the digitalized patterns of a fingerprint.

What is the use that NovaGenesis does for all these names? We asked these question many times while designing NG. The answer is: every information processing, storage, or exchanging depends on names and their relationships. The target of a communication is an unique name in a certain scope. The location of certain destination is also a name that provides the relative distance among possible targets. Ownership, equivalence, "is contained", and many other semantic operators can be represented by a *name binding* (NB). A NB can map several names to many other names/objects. Additionally, one can expect that people (and even machines in future) will denote other existences by names. Therefore, name bindings can represent the relationships (semantic operators) among named-existences. In this sense, a NB is itself another existence, a virtual/abstract one, which can be stored as a virtual object.

NovaGenesis is generic enough to enable the creation of any naming structure. A naming structure is an scheme to denote existences following some planned strategy. For example, in the current Internet hosts are denoted by an hierarchical name structure, where names have two portions: host and domain names, e.g., mycomputer.inatel.br. Using name bindings the CIA architects can design any naming convention. NG employs names to identify and locate communicating targets. All name bindings are stored in a distributed software forming a giant *name bindings graph* (NBG). Identification and location is a matter of scanning this graph to determine entities that belong to some scope or that inhabit some space. A communication target could be a content, a computer program, a computer, or any other existence.

3.2 Substrate Resources, Services, Contracts, Protocols, and Layers

Every software-based CIA (SB-CIA) is supported by physical world existences called *substrate resources*. Examples are antennas, fiber optics, microprocessors, memories, hard disks, etc. The exponential growth in computers capabilities is creating a phenomenon called *virtualization*. Maybe the most prominent example is the so called cloud computing where virtual machines (VMs) work like physical ones. More recently, virtualization on networking technologies is being addressed under the banner of network function virtualization (NFV) (Salsano et al., 2014). The idea is to replace customized hardware - many times deployed at difficult access sites - by software-implemented functionalities inhabiting VMs in the cloud. NG assumes that computing hardware is evolving so fast that software-based implementation of networking protocols is already possible for the majority of the network stack.

The increasing role of software in ICT architectures demand for excellence in software engineering. A technology for this purpose is to design software-as-a-service (SaaS). SaaS is often related to a service-

oriented architecture (SOA). We define a service as an existence aimed at processing, exchanging, or storing information. According to this definition, a computer program (or a process) is a *service*. Any substrate resource can be represented by named services, e.g., infrastructure-as-a-service (IaaS). Even protocol implementations provide services.

In this paper, no distinction is done between "protocol implementation" and a service. According to our service definition, a protocol is implemented as a service that processes, stores, and exchanges information in order to build networks. Thus, services use other services indefinitely, starting from the ones required to implement a network. This paper proposes the concept of *protocol-implemented-as-a-services* (PIaaS). Observe that an interface to expose any service to other services is required. NG enables SCNs to be used for this purpose.

NovaGenesis envisions a service life-cycle that includes features exposition, peer discovery, negotiation, contracting, monitoring, evaluation, and releasing. All these steps take advantage of the NBG. Service descriptors and contracts are named using SCNs. A contract is defined as a piece of information that sets the limits, responsibilities, clauses to be respected, as well as the criteria for completion and punishment of services that were poorly executed.

The concept of a *layer* still prevails in NG. A definition that combines the concepts of (Tanenbaum, 2003), OSI model (Standardization, 1996), (Day, 2008), and (Chaitin, 2010) is: a layer is an abstraction for a cluster of services that is offered in a distributed way to other services, isolating rules implementation, following a shared language - a common syntax and semantics - its interface. In this paper we replace the terms: "protocol implementations that are offered in a distributed way to other layers" by "services offered in a distributed way to other services". Thus, a NG layer is composed by several services that are exposed to other services via a predefined language.

3.3 Current Proof-of-Concept

A first implementation of the NBG and PIaaS concepts, covering intra node and inter node service communication was coded in 2012. We adopted a pub/sub model, where NBs and contents are published and subscribed by services. The NBG is implemented using distributed hash tables (DHT). NBs are published by services and stored on DHT. Service life cycling is build over this distributed pub/sub service. We implemented some protocols for service exposition, discovery, contracting, and named-content forwarding and routing. The following services have been designed for current version:

Hash Table Service (HTS) - It provides a domain level hash table that is used to store published NBs. Name bindings are categorized to improve scalability.

Generic Indirection Resolution Service (GIRS) - It forwards name bindings (together with content) for one or more HTS instances.

Publish/Subscribe Service (PSS) - It is the narrow waist for NG services. Any service will use the publish/subscribe directives provided by the PSS hierarchical service. Services can publish NBs and associated content to other services and subscribe other name bindings of their interest.

Proxy/Gateway Service (PGS) - To facilitate migration and enable transport over other technologies, we envisioned the PGS. The current PGS provides software-based messages encapsulation, forwarding, and routing. Additionally, the PGS is also a proxy for core NG services inside an operating system (OS). It represents these core services during bootstrapping, forwarding public NBs to other friend PGSs, exposing them to enable name-based self-organization. The PGS can maintain inter node IPC without TCP/IP only using Ethernet/Wi-Fi.

Application (App) - The current implementation has a generic application capable to explore the pub/sub service (PSS) offered by the core.

4 ADAPTING NovaGenesis: REQUIREMENTS AND CHALLENGES

4.1 Rethinking Naming

The proposed architecture shall rely on a strong naming approach to capture relationships among entities. Not only SCNs should be enabled for all existences, but also natural language names (NLNs) to enable in-architecture ontologies accommodation. NovaGenesis already supports natural language and/or self-certified name bindings to converge human and machine languages. This improves services expressiveness, like in current SOA, e.g., web services. People's names can be related to their equipment, services, and content as illustrated on Figure 1. Distributed name resolution enables services to explore other services, users, and content context and scope. NovaGenesis already implements this on PSS/GIRS/HTS services. NG generic naming structure can support UCN, CCN, ON, cloud requirements for naming.

4.2 Addressing Heterogeneity

A PGS can be implemented to offer gateway functionality to each technology deployed at the UCN, ON, CCN, etc. For example, if ZigBee technology is employed, a PGS can be implemented at the ZigBee gateway to bridge frames from/to NovaGenesis. The same can be done for every technology in the network. We are expanding PGS to include software-defined control functionalities. This new service will include proxy/gateway/controller (PGC) functionalities. It can expose non NovaGenesis devices features (and available configurations) to other NG services. Hence, after contract establishment, other services can publish configuration change requests that are translated to other technology devices, like ZigBee, Bluetooth, etc. This allows NG services to configure the network directly, creating what we are calling service-defined architecture (SDA).

4.3 Encouraging Collaboration

UCN provides the means for cloud and big data systems to be accessible to users, increasing their processing and storing capabilities. Users' devices can be enhanced by being given the chance of exploiting other users' devices in the vicinity, and being able to access the content that is made pervasively available at the user's current location. NovaGenesis SOA favors paid collaboration via dynamic SLA negotiation and establishment. Free models of collaboration are also possible. NG approach should be merged with our previous work towards a convergent approach. Previous work like SENSE (Ciobanu et al., 2014a) can help clarify which are the requirements for enhancing collaboration. SENSE is a collaborative selfish node detection and incentive mechanism for mobile networks where collaboration among users is a must. Since information collected locally by each node may not be sufficient to reach an informed decision, nodes running SENSE collaborate through gossiping. After informing each other of their observations, nodes reach decisions individually based on their local and received information. New NG services will implement a range of collaboration models, i.e. selfish, paid, and free. Spontaneity cluster formation based on user labeling can also be implemented. Services to determine node popularity are welcome.

4.4 Supporting Broad Opportunism

Our convergent architecture will require contextualized opportunities detection, which can be implemented as NG opportunity detection services (ODSs). These services can be fully integrated with NG PGCs to enable self-orchestration of exposed substrate resources. The PGC services expose physical resources (hardware) for software orchestration. Opportunities can include proximity, battery, offloading, spectrum, etc. All the required information to expose and explore opportunities is available for authorized services via PSS/GIRS/HTS. The architecture needs novel solutions for data aggregation. Since nodes running PGC services are generally small hand held devices such as smartphones, their memory is limited. Moreover, the access speed is also very important when there is a contact between nodes, since the duration of an encounter between two nodes (i.e. the time window when they can exchange data) is relatively short, due to the high degree of node mobility. Opportunities can be detected and shared using ODS. NG approach should be merged with previous work. Examples are ULOOP (http://copelabs.ulusofona.pt/˜uloop/), UCR (http://copelabs.ulusofona.pt/index.php/research/-projects/past-projects/151-ucr). In other words, a convergent name-based opportunistic routing/forwarding approach will be required. Regarding proximity, not only physical world contacts can be explored, but also service contracts. Opportunity notification can be done via PSS.

4.5 Seeing Everything-as-a-Service

NovaGenesis PGC services expose hardware resources to software allowing all physical resources to be seen as a service. Controllers, proxies, and gateways can be exposed as a services. All the required orchestration for our convergent architecture will be service-based. Service life-cycling is intrinsic, covering all aspects from exposition, discovery, negotiation, contracting, content exchanging, quality monitoring, and releasing. The goal is to accommodate protocol implementation as services, enabling them to dynamic establish SLAs, giving rise to networking self-organization based on detected opportunities. Flexible smart network services can discover each other, prepare SLAs proposals, negotiate with possible peers, establish SLAs, work together to explore social- and context-aware opportunities, evaluate partners, and finish SLAs.

4.6 Security, Privacy and Trust

The distributed scenario behind the proposed architecture poses several challenges regarding security, privacy, and trust. Networking cache, traffic offloading, opportunistic collaborations, and cloud offload-

ing are examples of user-centric approaches that will require new security models. NG employs a pub/sub communication model that favors the rendezvous among authenticated and authorized services. Content exchanging only happens after SLA establishment. Hence, it is secured by asymmetric cryptography. Publishers maintain a secure association with the PSS, which stores SCNs of authorized subscribing entities. Subscribers also have a secure association with PSS. The PSS only delivers the content after proper authentication and authorization. Additionally, the PSS provides revoking of published bindings, data, permissions, etc. The SLA-based self-organization enables the establishment of a trust network among peer services - which is ideal for UCN, ON, CCN, and clouds. All messages are confidential and have SCN-based integrity. We envision that NG will need new services for trust network formation, assertion, and management, as well as services for unbiased contract, reputation, and trust evaluation. NG name- and contract-based "social security" goes beyond traditional mechanisms.

4.7 User-Centric Life-Cycling

User-awareness and social-behavior awareness need to be estimated properly to drive NG ecosystem. The aim is to adapt protocol implementations according to user/social data. Hence, new services to estimate social trends and achieve context-awareness will be necessary. NG enables users to define high level policies that can be published to other services by a policy definition service (PDS). Published policies can be subscribed by peer services and used in their decision cycles. Big data and cloud information can be shared together with policies to make all the environment user-aware/socially-aware. For example, imagine a user agrees on selling part of this bandwidth to other users' radios when its device battery is charged more than 50%. This policy can feed network level protocols, in order to establish opportunistic routing among users devices. Use cases and proper policies should be designed and new NG services created to implement UCN and content life-cycling. Social engagement can be derived from available public information and explored by these emerging services. Interest-driven is favored published ontologies, helping on establishing successful partnerships.

4.8 Tolerating Delays and Disruption

Long delays and disruptive communication represent a challenge for current Internet stack. TCP does not fit well on long delays scenarios and UDP requires excessive application level programming. The asynchronous communication model provided by the PSS together with the HTS networking cache enable NG services to change information in different time moments. Connectivity disruption does not impact on sockets since a new naming structure is provided, turning names perennial, independent of connectivity. Protocols implemented as a service (PIaaS) offer the required flexibility to deal with intermittent connectivity and long delays. These PIaaSs need to be designed and implemented using NG software. A hardware implementation is also possible, but will require complete new designs using FPGA.

4.9 Supporting Mobility of Everything

What are the entities one expect to move in future user-centric architectures? The answer we propose is everything, from terminals, people, services, up to entire opportunistic networks. NG naming structure enables any name that satisfies some requirements to become an identifier or a locator. This decouples "what do you want" from "where it is". Mobility of everything is supported by rebinding names during movement. The identifier of what is moving remains the same - the only thing that changes are the locators. This solution relays on NG distributed NBs pub/sub and storage. UCN, CCN, NFV, and ON need mobility of everything support. NG can address this.

4.10 Integrating Cloud and Networking

Another important requirement is to alternate the use of computing and communication resources interchangeably. In other words, the lack of computing resources due to energy shortage or high load can be compensated by communication resources, which help on migrating the tasks to other machines. In contrary, when energy is limited (like in mobile devices), functions can be virtualized in the cloud (providing cloud offloading). Cloud networking is naturally supported. NG design should be merged with our previous work in cloud. Two approaches to be considered are: context-aware cloud computing infrastructure for taking good budgets. It is a SaaS that streamlines the interaction between for customers and sellers (salesman); and resource provisioning on cloud computing environments, which is an IaaS approach aimed to offer load balancing for a service that runs parallel applications. It consists in task migration considering the current pool of allocated processors. After that, if the SLA is not satisfied, our second approach will be to allocate new resources in order to run the application with the previous established requirements. Also,

it comprises the deallocation of resources if they are super estimated for running the application. Finally, the resources used by services can vary along their lifetime in accordance with the application behavior.

4.11 Providing Self-Organization

The scenario we are imagining requires new approaches for management and control. One can not expect people will manage or control dozens of devices connected to others, manually. The current management model depends on frequent human interference, having poor scalability when considering the swarms of devices we expect on next years. Embedded control functions (usually, at equipment control plane) create a complex distributed states solution, which challenges operators to keep a coherent and efficient network configuration. Future architectures need to self-organize according to user needs and detected opportunities. We envision a hierarchy of control loops that follow user-awareness and social behavior awareness to autonomically configure and manage ICT architectures. NG enables services to self-organize using NBs and content pub/sub. Self-management is fundamental in the proposed architecture, since no one will be responsible alone to manage a socially driven, possibly infrastructure-less architecture. Or previous work on UCN, ON, and CCN can help on designing these hierarchical control loops.

4.12 Control and Management

To address the requirements of effective utilization and optimization of heterogeneous resources (storage, processing, and networking), the architecture requires an innovative software-defined everything (SDE) paradigm. NG addresses this challenge by means of PGCs. A PGC represents some hardware resource in the software layer and can establish dynamic contracts in the name of them. It also controls the hardware devices, e.g., software-defined systems and/or radio, to perform the changes required. It can even expose hardware status to other NG services, enabling them to proactively prepare cloud/network solutions in advance to user requirements. It is a paradigm shift towards service-based network control and management. NovaGenesis's PGC services will represent all physical world resources used. Controllers- and managers-as-a-service will establish contracts with PGCs to create a new control/management model. Decision loops can be formed by establishing chains of control/management services linked via NG service contracts.

4.13 Supporting Context-awareness

User-, regulation-, situation-awareness, and many other context-awareness features are required to make sound decisions - decisions that consider the relevant contexts to every situation. NG enables services to securely and privately subscribe the relevant contexts following their trust network. In other words, the PSS provides a distributed networking cache from where services can subscribe the relevant contexts for decision making. For example, a radio resource manager could subscribe spectrum usage data from a spectrum analyzer service. The manager can form a logically centralized view of radio frequency spectrum usage at some location, the so called situation-awareness. Another example is related to social-awareness.The dynamism of social behavior can be derived from big data services implemented at NG or at any other software. In the latter case, a NG service will be required to bridge legacy software to NG cloud. Well established social trends can be published by a big data service (BDS) in order to feed other services.

4.14 Implementing Decision Cycles

The architecture shall adapt its behavior (cloud and networking aspects) according to users needs and detected opportunities, changing protocol implementations, data paths, parameters, etc. The architecture should explore dynamically the available connectivity, frequencies, bandwidths, technologies, and nearby friends capabilities. It must configure itself to take advantage of perceived opportunities in a reasonable time. In long term, autonomic and cognitive decision cycling will be required for self-management, auto-piloting, and opportunistic networking. Specialized services could implement required decision cycles. The cycle starts with user generated objectives, policies, rules, and regulations. An existing plan is selected. The plan is executed. The obtained results are collected and analyzed to measure the degree of success. If success was achieved, the objective is considered as met. Else, decision making can select changing the plan (or adapting it). This is an aim for future.

4.15 Addressing Society Challenges

Digital inclusion is one of the main aims of user-centric, opportunistic, infrastructure-less (or public infrastructures) architectures. It could be an important driver for developing countries like Brazil and India. Also, social-driven proposal have the potential to change our society towards "smart solutions", where every resource is better used, optimized for inclusion,

and green technologies requirements. We believe architectures like the one we are proposing on this paper have the important role of helping us to solve important social, economical, and environmental problems.

5 CONCLUSION

This paper proposed a convergent architecture that integrates emerging socially-driven, opportunistic, user-centric networking with NovaGenesis name-based, software-defined, information-centric, service-centric, self-organizing cloud networking proposal. Pre-requirements and open challenges regarding several topics have been discussed. NovaGenesis principles and current implementation provide a satisfactory substrate to implement the proposed architecture. NG joint orchestration of named-services and contents provides an appropriated environment to implement socially-driven/opportunistic/cloud/networking approaches as services. Even protocols are implemented as services, enabling the resultant architecture to react according to user-defined policies, rules, regulations, environment situations. Context-awareness can be included on decision making, changing protocol implementations according to social trends. The paper is a first step of an ongoing work that contributes to the community by discussing how to integrate so many relevant issues in only one architecture.

We envision that the proposed architecture can be implemented by: (i) specifying new NG services that meet the raised pre-requirements; (ii) adapting previous work techniques as new NG services; (iii) modifying NG core services accordingly; or (iv) integrating already existing software (without any modification) with NG via proxy/gateway/controller. This effort is expected to result into a convergent solution comprising the best of the considered architectures (ICN, DTN, UCN) and that allows users to seamlessly access content, and share resources anytime and anywhere in today's dynamic scenario over their powerful personal devices. Future work include NG performance, portability, and embedding on mobile devices.

ACKNOWLEDGEMENT

This work was partially supported by Finep/Funttel Grant No. 01.14.0231.00, under the Radiocommunication Reference Center (Centro de Referência em Radicomunicações- CRR) project of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações - Inatel), Brazil.

REFERENCES

Ahmed, S., Raza, A., Asghar, H., and Ghazia, U. (2010). Implementation of dynamic spectrum access using enhanced carrier sense multiple access in cognitive radio networks. In *Wireless Comm. Networking and Mob. Comp. (WiCOM), 2010 6th Int. Conf. on*, pages 1 –4.

Alberti, A., de O Fernandes, V., Casaroli, M., de Oliveira, L., Pedroso, F., and Singh, D. (2014). A novagenesis proxy/gateway/controller for openflow software defined networks. In *Network and Service Management (CNSM), 2014 10th Conf. on*, pages 394–399.

Alberti, A. M. (2012). Searching for synergies among future internet ingredients. In Lee, G., Howard, D., Slezak, D., and Hong, Y., editors, *Convergence and Hybrid Info. Technology*, v. 310 of *Comm. in Comp. and Information Science*, pages 61–68. Springer.

Caini, C., Cornice, P., Firrincieli, R., and Lacamera, D. (2008). A dtn approach to satellite communications. *Selected Areas in Comm., IEEE Journal on*, 26(5):820–827.

Chaitin, G. (2010). *Mathematics, Complexity and Philosophy*. Midas.

Ciobanu, R.-I., Dobre, C., and Cristea, V. (2013). Sprint: social prediction-based opportunistic routing. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 14th Int. Symp. on a*, pages 1–7. IEEE.

Ciobanu, R.-I., Dobre, C., Dascălu, M., Trăuşan-Matu, Ş., and Cristea, V. (2014a). Sense: A collaborative selfish node detection and incentive mechanism for opportunistic networks. *Journal of Network and Comp. App.*, 41:240–249.

Ciobanu, R.-I., Marin, R.-C., Dobre, C., Cristea, V., and Mavromoustakis, C. X. (2014b). Onside: Socially-aware and interest-based dissemination in opportunistic networks. In *Network Operations and Management Symp. (NOMS), 2014 IEEE*, pages 1–6. IEEE.

Conti, J. P. (2006). The internet of things. *Communications Engineer*, Vol 4, 2006.

Day, J. (2008). *Patterns in Network Architecture: A Return to Fundamentals*. Prentice Hall.

Jamshidi, P., Ahmad, A., and Pahl, C. (2014). Autonomic resource provisioning for cloud-based software. In *Symp. on Software Eng. for Adaptive and Self-Managing Systems*, SEAMS 2014, pages 95–104, ACM.

McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.

Moreira, W. and Mendes, P. (2013). Social-aware opportunistic routing: The new trend. In Woungang, I., Dhurandher, S. K., Anpalagan, A., and Vasilakos, A. V., editors, *Routing in Opportunistic Networks*, pages 27–68. Springer.

Moreira, W. and Mendes, P. (2015a). Dynamics of social-aware pervasive networks. In *Int. Workshop on the Impact of Human Mobility in Pervasive Syst. and App., 2015 (PerMoby'15)*.

Moreira, W. and Mendes, P. (2015b). Impact of human behavior on social opportunistic forwarding. *Ad Hoc Networks*, 25, Part B(0):293 – 302. New Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Arch., Infrast., and Comm..

Moreira, W., Mendes, P., and Sargento, S. (2012). Opportunistic routing based on daily routines. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE Int. Symp. on a*, pages 1–6.

Moreira, W., Mendes, P., and Sargento, S. (2014). Social-aware opportunistic routing protocol based on users interactions and interests. In Sherif, M. H., Mellouk, A., Li, J., and Bellavista, P., editors, *Ad Hoc Networks*, v. 129 of *Lecture Notes of the Institute for Comp. Sciences and Telecomm. Eng.*, pages 100–115. Springer.

Papazoglou, M., Traverso, P., Dustdar, S., and Leymann, F. (2007). Service-oriented computing: State of the art and research challenges. *Computer*, 40(11):38 –45.

Salsano, S., Blefari-Melazzi, N., Presti, F., Siracusano, G., and Ventre, P. (2014). Generalized virtual networking: An enabler for service centric networking and network function virtualization. In *Telecommunications Network Strategy and Planning Symp. (Networks), 2014 16th Int.*, pages 1–7.

Sofia, R. and Mendes, P. (2008). User-provided networks: consumer as provider. *Communications Magazine, IEEE*, 46(12):86–91.

Sofia, R., Mendes, P., and Moreira, Waldir, J. (2014). User-centric networking: Living-examples and challenges ahead. In Aldini, A. and Bogliolo, A., editors, *User-Centric Networking*, Lecture Notes in Social Networks, pages 25–51. Springer.

Standardization, I. (1996). Iso/iec 7498-1: 1994 information technology-OSI basic reference model. *Int. Standard ISOIEC*, 74981:59.

Tanenbaum, A. S. (2003). Computer networks, 4-th edition.

Xylomenos, G., Ververidis, C., Siris, V., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K., and Polyzos, G. (2014). A survey of information-centric networking research. *Comm. Surveys Tutorials, IEEE*, 16(2):1024–1049.

Simulation as a Service
A Case Study of Provisioning Scientific Simulation Software via a Cloud Service

Morgan Eldred, Alice Good and Carl Adams
School of Computing, University of Portsmouth, Portsmouth, U.K.
morgan.eldred@myport.ac.uk, {alice.good, carl.adams}@port.ac.uk

Keywords: Cloud Computing, Simulation as a Service, Software as a Service (SaaS).

Abstract: This paper reports on a case study that was conducted on a large scale cloud service project that moved scientific simulation software to the cloud, one that used sensitive data. The study aimed to explore the challenges and practicalities of initiating and evaluating simulation as a cloud service. Action research was used to examine the nuances throughout the project as the service was moved from on-premise into a public cloud, lasting over one year from start to finish. The study was able to identify some emergent issues affecting initiation, technical security challenges and the evaluation of a significant change in a critical applications provisioning model.

1 INTRODUCTION

During the last 20 years there has been a continuing trend towards IT industrialisation. This has resulted in IT services becoming repeatable and usable by a wide range of customers and service providers. This is because of the increasing commoditization of technologies, virtualization and the rise of service-oriented software architectures, along with the dramatic growth in use of the Internet. These factors constitute the basis of a discontinuity that offers opportunities to shape the relationship between those who consume and those who provide IT services. The discontinuity implies that the ability to deliver specialized services in IT can now be paired with the ability to deliver those services in an industrialized and pervasive way. The reality of this implication is that users of IT services can focus on the business capability of what the services provide, rather than how the services are implemented or hosted. Similar in nature to how utility companies sell power on demand to subscribers, IT services can now easily be delivered an provisioned as a contractual service. This is not a new concept, but it does represent a different model from the licensed-based, on-premises models that have traditionally dominated the IT industry.

Cloud services provide a new way of delivering computing resources. Several types of cloud computing platforms exist, of which the main types are public, private and hybrid. Public clouds are normally offered by commercial organisations that provide access for a fee. Private clouds exist within are contained within a specific organisation and typically are not available for outside use. Hybrid clouds are a mixture of private and public clouds with the typical setup being that of a private cloud that has the ability to call upon additional resources from a public cloud (Chang, 2014).

The main advantage of cloud computing is the ability of equilibrating the access to computing resources for all types of businesses, regardless their dimensions and investment capabilities. These advantages include cost efficiency, scalability, concentration, security and accessibility with a further list below.

This paper outlines the overview, key issues and themes that emerged in a study of a large scale project within a mid-sized multinational company that ran a pilot to provision a scientific simulation software package via a public cloud.

2 RESEARCH METHODOLOGY

The research was conducted via a case study, taking an action research approach which used an iterative approach to collecting and analysing data. The benefits of an action research approach are that it focuses on generating solutions to practical problems and empowers the researcher to engage with the research and subsequent implementation

activities (Mayer, 2000). A typical action research methodology takes a five step approach, as follows:

- Step 1: Identify the Problem
- Step 2: Devise a Plan
- Step 3: Act to Implement a Plan
- Step 4: Observe
- Step 5: Reflect and Share

Using this methodology, the approach starts with identifying the problem, which in this case was to determine if the simulation software was able to run via a cloud service. The second step was then to devise a plan around the migration of the service to the cloud and then test the success criteria. The next step was to execute the plan and implement the service, via a cloud provisioning model. This is the part of the approach where the action research is taking place via an iterative approach. After the plan was implemented, the researcher would observe how the service was or was not working. Once the researcher has had time to observe the situation then the entire process of action research was reflected upon, and at times the whole research approach may start over again (McCallister, 2011).

For this research, the researcher was a participant observer who was present for: top management meetings; from the inception of project to start-up; to designing the service; all the way throughout the whole project, till the end state of deciding if the service would be provisioned via the cloud. This access provided rare insight into what goes on in a multinational organisation during a large scale cloud service project. Along with access to top management meetings, the researcher had access to the critical role of the Head of IT strategy & projects, which was the primary role for orchestrating one of the biggest cloud pilots within the industry. The research itself used an academic approach to a real-world case study.

During the observation and reflection stages of the action research approach, mixed methods were used to evaluate the success of the project. These included quantitative methods that were used to determine the technical success criteria. These methods looked at indicating whether the data would be consistent before migrating to the cloud and then in determining the run-time performance of the simulations within a cloud environment.

The researcher used a mixed method of both qualitative and quantitative data, in the form of surveys. These were distributed to twenty four employees of both technical and business staff to find insight into trends that occur and organisational challenges. The use of a mixed method helped back up one set of findings from one method of data collection underpinned by one methodology, with another different method underpinned by another methodology. The researcher designed a series of survey questionnaires that included both boolean and open-ended questions, so that the resulting data would be both qualitative and quantitative. Qualitative data was used and analysed in the following approach. Questionnaires with open-ended questions were sent to twenty four pre-selected participants, coming from a wide range of both technical and business staff. The Questionnaires were distributed electronically via an online survey tool, with replies sorted and trends were identified to find commonalities. Upon the initial analysis another set of quantitative questionnaires was distributed to further investigate the findings and commonalities. The decision on the selection of interviewees was determined via a deductive approach to responses.

Examples of the specific questions that were used in the survey are listed below.

- Was the project a success
- Is cloud a viable provision model for scientific applications
- Is cloud scalable to run simulations
- Was the organization ready for cloud
- Does the organization need to introduce new processes for the adoption of cloud

Examples of the specific open-ended questions that were asked are listed below.

- What emerging themes were identified
- What key issues that were identified
- What was the impact of key issues

3 CASE STUDY

The case study is based on a midsized international company with a headquarters in Europe with operations in Europe, the Middle-east and Africa. The company has approximately 5,000 employees located over seven countries with revenues consistently averaging between $8-10 billion dollars and has a corporate culture that promotes innovation.

The company was exploring the possibility of migrating scientific simulation software that has significant computing and storage requirements into a cloud based HPC environment service delivery model. The benefits of moving from on-premise to a cloud provisioning model were that it would enable the scientific community within the company to flexibly increase compute via a cost effective, on-demand, pay-per use model (Jackson et al, 2012).

If successful, this new capability would enable the company to compete with much larger companies who had the capital to invest in the development and maintenance of large scale on-premise super-computing environments. A project was conducted to do a formal evaluation of migrating the service to the cloud and to determine if the concept was feasible from a technical and economic perspective, before a decision to invest further into a cloud provisioning service model was decided. The project was a multimillion dollar project, lasting over a year that consisted of a five person project team, with twelve other stakeholders from IT and outside IT whom were involved in the project.

3.1 Problem Statement

Scientists within the organisation within this case study were being challenged with a need for superior simulation modelling, as both the supply of information and the sophistication of quantitative techniques increases. The organisation invested heavily in technology that providing a vastly higher resolution of raw data, generating unprecedented volumes of data. All this additional data enables finer-scale simulation, as the geo-cellular models they simulated burst through the 10 and 100 million cell thresholds. As impressive as these advances were, they only represent more granular approaches to traditional modeling methods.

As the models increase in size, the organisation requires significantly more computing resources to run, given the increasing complexity with detailed models needing to run hundreds of times to quantify uncertainties and define the risks. The growth in demand for high performance computing was exceeding the supply from vendors. This results in the organisations' science community needing to limit simulations due to computing capacity, and was the driving force in running a proof of concept to explore new sources of high-performance computing capacity via a cloud provisioning model.

With the organisation investing in the project, it would need to determine real practical questions in relation to simulation as a service such as:

- Will the project be successful
- Is cloud a viable model for scientific applications
- Is cloud scalable for simulations
- Is the organisation ready for cloud
- Will new processes need to be implemented for cloud
- What themes would emerge
- What key issues would be faced

3.2 Plan

The project plan was drafted during the development of the business case. The project was broken down into six milestones, with each milestone having an estimated time. The milestones consisted of having: a signed off business case; a contract agreement with vendors; a high and low level design; the implementation; testing and finally an analysis conducted on the results leading to the project findings. Overall the initial estimation of the project plan was that it would take around five months to complete, however the actual time for the project was fifteen months. Significant challenges were faced in almost every step of the project plan.

Business Case: This stage took four months instead of an expected one month, as the biggest delay was in getting approval and buy-in for the business case, with the critical element revolving around the way that the sensitive data used within the simulations would be protected.

Contract: Similar to attaining business case approval, getting a contract signed with the software vendor took four months instead of the expected one month. This was because neither the organisation nor the software vendor could come to an agreement around intellectual property rights. In the end the situation was resolved as both parties agreed to waive rights.

Design & Implementation: The design took twice as long as expected, due to stringent internal information and data security requirements. This impacted the implementation as the vendor software required using a physical license dongle. As the project took an action research approach, the design and implementation phases took an iterative approach and required reworking of the designs during the implementation phase.

Testing of the service was approximately three

Table I: Milestones.

Milestone	Timelines	
	Estimated	***Actual***
Business Case	1 Month	4 Months
Contract Agreement	1 Month	4 Months
Design	6 Weeks	3 Month
Implementation	1 Week	1 Month
Testing	3 Months	3 Months
Findings	1 Month	2 Months

months the same time as initially estimated.

Findings: The findings took two months instead of the estimated one month, as the key themes and issues, led to some insightful findings, which required further investigation into finding commonalities and in conducting in depth interviews to validate the findings.

3.3 Design

The design was developed with the guiding principle that the architecture would need to be secure, lean and agile. This is because it was hypothesised that it would drive efficiencies and reliability through an elastic architecture that could dynamically scale up or down compute clusters as needed. The objective of the design was to simulate a real-life corporate network within a cloud scenario, so that the cloud service would be almost identical to the on-premise service, so that data could easily be moved in and out. As such a Virtual Private Cloud-VPC in the Amazon European datacentre was setup to act as the "corporate network". The next step was to create a VPC in an Amazon US datacentre to act as the "cloud network". The connections between the installations were facilitated through the use of OpenVPN which was installed on standard Linux Amazon Machine Images.

A major design requirement for running simulations in the cloud involves how to transfer large datasets between the corporate network and the cloud environment. To achieve this, a cloud network attached storage-NAS server was provisioned in the cloud, with a virtual device in the Amazon cloud configured to acts as a NAS front end to Amazon's object based data cloud, simple storage service-S3. Due to the large storage requirements; there was a need for a common internet file system, which is a standard way that computer users share files across corporate intranets and the internet, with a network file system interface. This design is commonly known as a cloud storage security gateway system and is considered a secure way for encrypting and decrypting data as it is either uploaded or downloaded via Amazon's S3 by examining the consistency of the contents and preventing data tampering (Wang et al 2013).

The next design step was to create the Simulator software head node in the cloud. This node would be static and would be were the simulation jobs would be submitted to and then run in a dynamically-created compute cluster. The head node rand on a red hat enterprise linux node running on an Amazon C3.xlarge size server. This server was selected as it had the required compute capacity need for the simulations along with having a solid state drive and a 10GB network interface. The design choice for using a solid state drive is that they offer higher performance compared to traditional storage devices and are needed in HPC systems, especially those with a growing demand of supporting big data applications (Chen et al, 2013).

A major security challenge was in the need to connect the simulation software's physical Universal Serial Bus-USB license dongle to a virtual server. This was resolved via the use of a USB network device server being placed within a de-militarised zone-DMZ within the corporate network. This enabled the mapping of a USB port to a virtual server over the network. The USB port on the device server was then mapped to the simulator license server in the DMZ and was configured with a public IP. The DMZ was also configured to allow traffic between Amazon and the license server.

Figure 1 depicts the conceptual design which indicates the three main networks within the setup, the Amazon cloud, the virtual office and the DMZ and the major components within each of these networks.

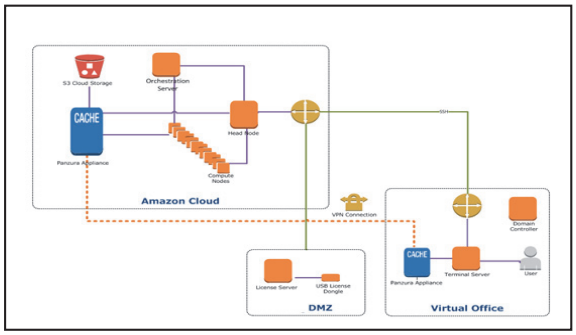

Figure 1: Conceptual Design.

A key component in the design was the NAS secure storage gateway (which also acted as a cache). Figure 2 depicts the dataflow from the simulation cloud which was within the amazon cloud. The data was de/encrypted as it passed through the cloud NAS and resided in the amazon storage, until it was to the NAS in the virtual office cloud, where it was then de/encrypted and passed into the main data source.

Along with the data flow there was a need to have a connection into the corporate DMZ as the simulation license server needing to be on the same subnet as the USB device server, thus not enabling the license server to be placed in the Amazon Cloud. The detailed design for this is depicted in Figure 3.

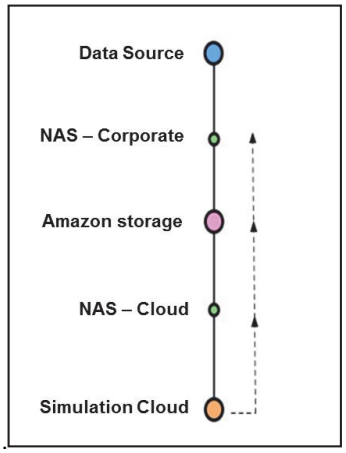

Figure 2: Data Flow.

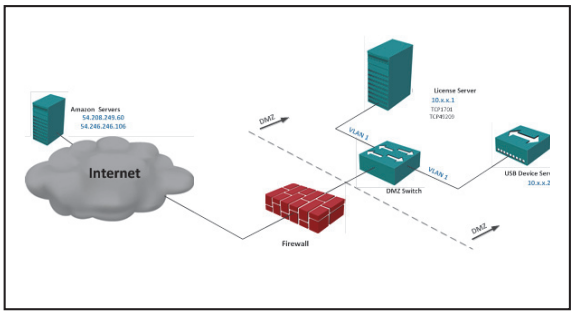

Figure 3: DMZ Design.

3.4 Technical Validation

Before the simulation cloud service was provisioned, an on-premise validation test was needed to ensure that simulations would run with the same accuracy regardless of the technology provisioning model. The approach taken for testing was to test out the characteristics with different stakeholders, to ensure requirements from users and those supporting the simulation software were properly gathered. The guidance provided was that the validation would need to ensure that the business characteristics, such as the need to ensure consistency when running simulation on the scalable dataset. The technical characteristics were that cpu performance scalability would need to be performed.

To achieve this, four cases were run on a workstation and then moved to the companies on-premise cluster which had a maximum of 8 core cpu's. The test ran the cases on the on-premise cluster using one, four and 8 cpu's to ensure consistency. The results from this indicated that moving the simulation jobs, did not have an impact on the jobs and once this was successful the simulation jobs could be migrated to the cloud service. Figure 4 indicates that the four cases demonstrated identical results for production rate and cumulative production for the duration of the field history as expected:

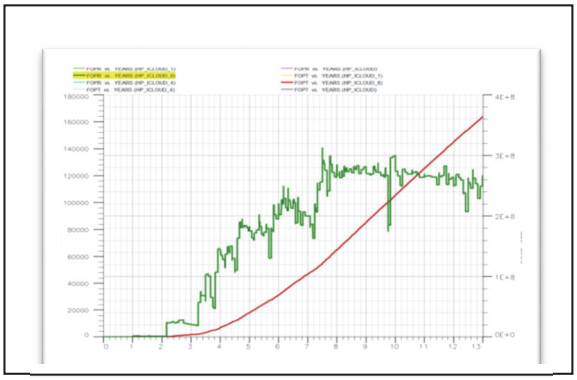

Figure 4: Simulation Validation.

The case with a single CPU on a workstation was completed in 20.9 hours, while the case on the on-premise cluster was completed in 19.6 hrs. The run time for the on-premise cluster with four and eight 8 CPUs ran at 7.7 and 5.7 hours respectively. The in-house cluster setup at the time of this study did not allow job executions on more than 8 CPUs. Figure 5 demonstrates the relationship between number of CPUs and wall clock time.

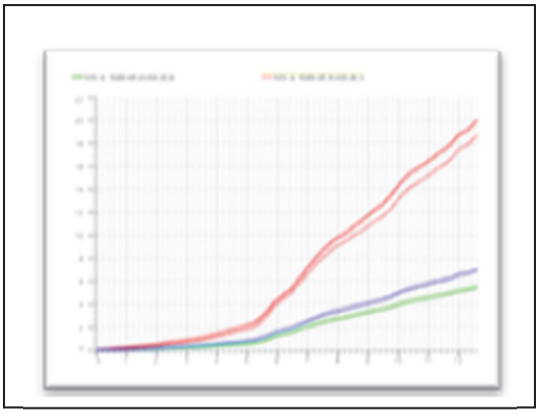

Figure 5: CPU Performance Validation.

With the run simulations being validated and with the performance of the on-premise service being measured, the next step was to compare this against the results of the performance within the cloud provisioned serviced.

Figure 6 shows the results, and indicates that the wall clock time decreases as more CPUs were added, for both calculations for the on-premise and cloud service. It was observed that on-premise calculations stagnated at more than 4 CPUs,

resulting in sublinear scaling. This is contrary to the performance via the cloud service, which observed close to linear scaling. Assuming that this linear scaling persists when adding more than 8 CPUs, extrapolating from this observation, it was hypothesised that for larger jobs, the performance of the cloud provisioned service would significantly increase compared to what can be achieved on-premise. This analysis is not exhaustive, but was severely limited by the size of the on-premise service having only 8 CPUs.

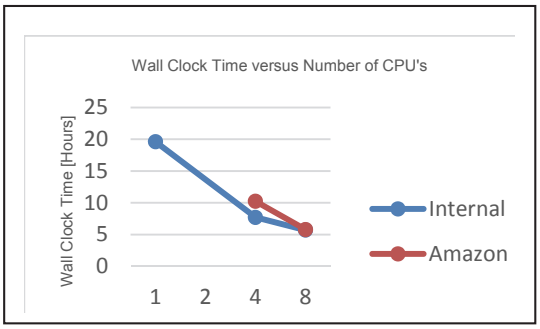

Figure 6: On-premise & Cloud Performance Validation.

3.5 Data Collection & Analysis

Once the system was implemented and had passed the technical validation aspect, twenty four individuals involved in the project completed a questionnaire, with 15 coming from IT and 9 from outside.

The 24 interviewees were asked: whether they thought the project was a success; whether cloud was a viable service model for scientific applications and if cloud was a scalable to run simulations; if the organisation was ready for cloud and if the organisation needed to introduce new processes for cloud. For data analysis purposes yes equating to a score of 1, while a no equated to a score of 0. The response and standard deviation were calculated as indicated in Table 1. Not all questions had an input, as respondents preferred not to say and the breakdown is as follows:

- Question 2: 4 participants declined
- Question 3: 1 participant declined
- Question 4: 1 participant declined

The following key insights were indicated

- 80% indicated that the project was a success.
- 71% indicating that cloud was a viable service model for scientific applications.
- 79% indicating that cloud is a scalable.
- Surprisingly only 25% indicated that the organisation was ready for cloud.
- 71% indicating that the organisation needed to implement new processes for cloud.

Table 1: Success Criteria.

Question	Response		Mean	Standard Deviation
	Yes & (Total %)	No		
1. Was the project a success	20 (80%)	4	**0.833**	**0.381**
2. Is cloud viable for scientific applications	17 (71%)	3	**0.850**	**0.366**
3. Is cloud scalable	19 (79%)	4	**0.826**	**0.388**
4. Is the organisation ready for cloud	6 (25%)	17	**0.261**	**0.449**
5. The organisation needs new processes for cloud adoption	17 (71%)	7	**0.708**	**0.464**

The same interviewees were then asked their opinion of the three major themes which emerged during the life of the project.

The following key insights were indicated

- 62.5% indicated that politics were a prevalent theme, 38% indicated that politics was the first theme, 25% indicated it was the second theme.
- Innovation was second with 45.8%, 25% for the first theme, 8% for the second and 13% for the third.
- Security at 33.3%, 13% for the first, 17% for the second and 4% for the third.

Other key themes included vendor solutions, intellectual property rights, a lack of the required skills, internal processes, business value and change.

Interviewees were then asked their opinion of the three major issues which emerged during the life of the project.

The following key insights were indicated

- Politics was again the highest at 66.7% which was aligned with the responses from the emerging themes, with 42% of respondents indicating it was the first issues while 25% indicated it was the second issues and similar to themes zero respondents indicated that it was the third issue.

Table 2: Emerging Themes.

Theme	First	Second	Third	Inclusion Total %
Business Value	1	2	0	12.5%
Change	0	2	1	12.5%
Innovation	6	2	3	45.8%
Intellectual Property Rights	2	2	1	20.8%
Politics	9	6	0	62.5%
Processes	0	2	1	12.5%
Security	3	4	1	33.3%
Skills	0	1	2	12.5%
Vendor Solutions	3	1	1	20.8%

- Project Management was second with 37.5%, 21% for the first theme and 16.5% for the second.
- Contracts and Processes were tied for the third issue both with a response of 20.8% with contracts had a response of 12.5% for being the first issue and 4.15% for being the second and third issues.
- 4.15% identified processes as being the first issue and 16.65% for being the third issue.

Other key issues included capability of staff, lack of clear KPI's to measure and information Security.

Table 3: Key Issues.

Key Issue	First	Second	Third	Total	Total %
Capability	2	2	0	4	16.7%
Contracts	3	1	1	5	20.8%
KPI's	0	1	1	2	8.3%
Politics	10	6	0	16	66.7%
Processes	1	0	4	5	20.8%
Project Management	5	4	0	9	37.5%
Security	0	0	1	1	4.2%

4 DISCUSSION

The pilot was initially resisted by internal members of the IT department that were responsible for supporting the simulation software. This delayed the approval of the business case. Design challenges arose during the design and implementation stages due to stringent internal security requirements.

This was a surprising finding as the respondents did not indicate information security to be an issue, this along with the data captured from the surveys indicated that politics was the pervasive theme and key issue of moving to the cloud. Considering that the project was determined a success, but that the organisation was not ready and would need to introduce new processes to support cloud. This leads to a finding of how organisation behavior and the perception of trust in security pose a real threat to the adoption of cloud. This is indicated by 2009 Gartner survey with indicates that politics is a challenge of cloud service adoption (Gartner, 2009).

Resistance to change is a normal human response as employees seek to translate the change to a personal context, which can be greatly magnified by fear of the unknown (Berube, 2012). If internal policies and security concerns are a significant challenge in cloud service adoption, then building a maturity assessment at the start of the project to understand the organisations culture, internal processes would clearly assist in migration to a cloud service, and the delivery of a cloud service project could then plan accordingly to further train and develop staff on the impact of cloud, before any implementation would occur. This activity was not included in the project, but the insights received after the fact, could help in any future cloud project by being able to measure and mitigate the risks of the cloud.

The data which was collected using an action research approach indicates that a lot is still unknown about dealing with challenges during the initiation stages of a cloud project were the realization that the change from one modal of working to another different modal has a significant impact on the success of a project. Though the project was validated as being a success, several emergent themes impacted the adoption. One significant emergent theme from the research was that the organisation did not have the appropriate internal policies.

The research shows that the evaluation and adoption of simulation as a service project, which is a considerable change to business practices, will likely involve more than technical performance and business improvements: It will also need to consider the wider political fault-lines of issues that would impact the acceptance from various stakeholders.

5 CONCLUSIONS

Cloud computing is maturing, but there is still a lot that remains uncertain for its adoption within

enterprises, such as the organizational changes brought about by cloud computing. Cloud services that support simulation via a HPC environment are attracting more attention in literature, in big business and in governments.

This paper has reported on research exploring the practicalities of conducting a significant simulation as a service project within a large company. This paper further explores the practicalities and contexts the issues of applying cloud to larger compute processing needs

This is one of the few works that covers simulation as a service in a real life project.

The research involved an iterative methodology based upon an action research methodology and covered all the stages of the project from creation to evaluation. The pilot project and research focused on evaluating the possibility of running simulation as a service which leverage a cloud infrastructure to address the HPC needs of the multinational company using a range of criteria, including technical capability and wider business case.

It was a successful project and the insights taken from this work can further be used to make informed decisions about moving simulations to the cloud. Lessons learned from this would be that doing a proof of concept is a good method.

The data which was collected using an action research approach indicates that a lot is still unknown about dealing with challenges during the initiation stages of a cloud project were the realization that the change from one modal of working to another different modal has a significant impact on the success of a project. Though the project was validated as being a success, several emergent themes impacted the adoption. One significant emergent theme from the research was that the organisation did not have the appropriate internal policies

The research shows that the evaluation and adoption of simulation as a service project, which is a considerable change to business practices, will likely involve more than technical performance and business improvements: It will also need to consider the wider political fault-lines of issues that would impact the acceptance from various stakeholders.

Developers and project managers can take practical guidelines from this paper that can be used to make informed decisions about moving simulations to the cloud. These examples are in the form of design, validation steps but more importantly the need to get feedback from different stakeholders before starting a project and the need to have an understanding of the potential political impact may occur similar to this project in terms of project delays and in design requirements. Key contributions to knowledge are that even if the project is successful, the organisation may not be ready for cloud and that new processes would need to be developed to operate via a cloud provisioning model. For considerably sized projects of this type the recommendation is to run a pilot first and to plan and execute the development of internal processes that are required to enable the organisation to be cloud ready.

REFERENCES

H. F. Wang, L. J. Wang, Pingjian Institute of Information Engineering, Beijing, China, *Chinese Academy of Sciences*, 2013.

N. Agrawal, V. Prabhakaran, T. Wobber, J.D. Davis, M. Manasse, R. Panigrahy, *Design Tradeoffs for SSD Performance*, Microsoft Research, Silicon Valley.

J. Chen, Using pattern-models to guide SSD deployment for Big Data applications in HPC systems, *2013 IEEE International Conference on Big Data,* 2013.

V. Chang, The Business Intelligence As a Service in the Cloud, 37, 512-534 ed. , *Future Generation Computer Systems*, 2014.

C. Vecchiola, S. Pandey, R. Buyya, High-Performance Cloud Computing: A View of Scientific Applications, Pervasive Systems, Algorithms, and Networks, *10th International Symposium on Pervasive Systems, and Networks*, 2009.

V. Chang C S. Li, D. De Roure, G. Wills, R. Walters, C. Chee, The Financial Clouds Review, 1 (2). pp. 41-63. ISSN 2156-1834, eISSN 2156-1826. ed. , *International Journal of Cloud Applications and Computing*, 2011.

M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, *A View of Cloud Computing*, Vol. 53 No. 4 ed. , ACM, 2010.

J. Meyer, Using qualitative methods in health related action research, 320: 178-181 ed. , *British Medical Journal,* 2000.

J. McCallister, *Contemporary Social Work Issues*, MSW, 2011.

K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud, *IEEE Second International Conference on Cloud Computing Technology and Science*, 2010.

Gartner, Managing and Deploying Clouds, *Gartner 2009 Datacenter Conference,* 2009.

D. Berube, *Resistance to Change is a Good Thing*, Life Cycle Engineering, 2012.

A. Khajeh-Hosseini, I. Sommerville, I. Sriram, Research Challenges for Enterprise Cloud Computing, *1st ACM Symposium on Cloud Computing,* 2010.

Disaggregated Architecture for at Scale Computing

Chung-Sheng Li[1], Hubertus Franke[1], Colin Parris[2] and Victor Chang[3]
[1]*IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A.*
[2]*GE Global Research Center, One Research Circle, Niskayuna, NY 12309, U.S.A.*
[3]*School of Computing, Creative Technologies and Engineering, Leeds Beckett University, Leeds LS6 3QS, U.K.*
{csli, frankh}@us.ibm.com, colin.parris@ge.com, v.i.chang@leedsbeckett.ac.uk

Keywords: Disaggregated Datacenter Architecture, Software Defined Environments, Software Defined Networking.

Abstract: The rapid growth of cloud computing both in terms of the spectrum and volume of cloud workloads brought many challenges to the traditional data center design - fast changing system configuration requirements due to workload constraints, varying innovation cycles of system components, and maximal sharing of systems and subsystems. In this paper, we developed a qualitative assessment of the opportunites and challenges for leveraging disaggregated datacenter architecture to address these challenges. In particular, we compare and contrast the programming models that can be used to access the disaggregated resources, and developed the implications for the network and resource provisioning and management.

1 INTRODUCTION

Cloud computing is quickly becoming the fastest growing platform for deploying enterprise, social, mobile, and analytic workloads. As many of these workloads grew to internet scale, they have ushered a new era of datacenter scale computing on top of the previous centralized and distributed computing era (Barroso 2013). During the *centralized computing era*, the computing resources are fully shared (share everything) and centralized managed. The subsequent *distributed computing era* allows decentralized management of distributed resources interconnected by networks. The "*at scale" computing era*, in contrast, involves *de facto* centralized management of massive amount of distributed and often virtualized resources that are locally concentrated within mega-datacenters and often spread across multiple datacenters. Recently, the need for increased agility and flexibility has accelerated the introduction of software defined environments (which include software defined networking, storage, and compute) where the control and management planes of these resources are decoupled from the data planes so that they are no longer vertically integrated as in traditional compute, storage or switch systems and can be deployed anywhere within a datacenter (Li et al., 2014).

The emerging at scale cloud data centers are facing the following challenges: fast changing system configuration requirements due to highly dynamic workload constraints, varying innovation cycles of system components, and the need for maximal sharing of systems and subsystems resources in order to minimize the Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) for efficient at scale operation. These challenges are further elaborated below. Enabling software and platform as a service with optimal stack level cost performance has become a major differentiator in the marketplace, especially for those services involving massive scale out environment such as Hadoop and Spark.

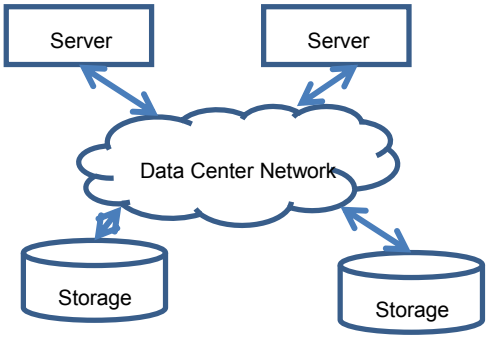

Figure 1: Traditional datacenter with servers and storage interconnected by datacenter networks.

Systems in a cloud computing environment often have to be configured differently in response to different workload requirements. A traditional datacenter, as shown in Fig. 1, includes servers and storage interconnected by datacenter networks. A

typical server system configured with only CPU and memory while keeping the storage subsystem (which also includes the storage controller and storage cache) remote is likely to be applicable to workloads which do not require large I/O bandwidth and will only need to use the storage occasionally. This configuration is usually inexpensive and versatile - but unlikely to perform well when large I/O bandwidth or small latency become pertinent. Alternatively, the server can be configured with large amount of local memory, SSD, and storage. This configuration, however, is likely to become expensive. Some of the system resources such as the SSDs configured within the server could be potentially underutilized at various times as workloads may not always be able to fully utilize them.

Traditional systems also impose identical lifecycle for every component inside the system. As a result, all of the components within a system (whether it is a server, storage, or switches) are replaced or upgraded at the same time. The "synchronous" nature of replacing the whole system at the same time prevents earlier adoption of newer technology at the component level, whether it is memory, SSD, GPU, or FPGA. The average refresh cycle of CPUs is approximately 3-4 years, HDDs and fans are around 5 years, battery backup (i.e. UPS), RAM, and power supply are around 6 years. Other components in a data center typically have a lifetime of 10 years. Consequently, an integrated system with CPU, memory, GPU, power supply, fan, RAM, HDD, SSD likely has the same lifecycle for everything within the system as replacing these components individually will be uneconomical.

Traditional system resources (memory, storage, and accelerators) configured for high throughput or low latency usually could not share these resources across the data center as they are only accessible within the "system" they are in. As a result, the utilization of the resources could be low. Those resources configured as remote (or network attached) allow maximal shareability but the performance in terms of throughput and latency are usually poor. As an example, challenges in terms of operational efficiency and security in the cloud based financial service domain were reported in Chang (2014), where a pipelined cloud service architecture is implemented on top of a traditional datacenter architecture.

In this paper, we developed a qualitative assessment of the approaches and challenges for leveraging the disaggregated architecture for at scale cloud datacenters. We compare and contrast the programming models that can be used to access the disaggregated resources. We also developed the implications for the network and resource provisioning and management. Based on this qualitative assessment and early experimental results, we concluded that disaggregated architecture with appropriate programming models and resource provisioning is likely to achieve improved datacenter operating efficiency. This architecture is particularly suitable for heterogeneous workload environments as these environments often have dynamic resource requirements and can benefit from the improved elasticity of the physical resource pooling offered by the disaggregated datacentre architecture.

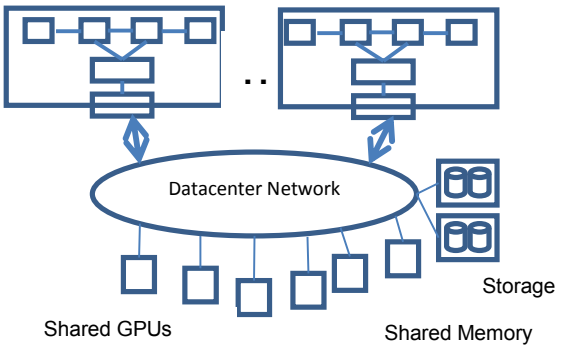

Figure 2: Partially disaggregated datacentre with both fully integrated server /storage as well as disaggregated GPU, SSD, and memory pools.

2 COMPOSABLE SYSTEMS ARCHITECTURE

The emerging disaggregated datacenter architecture or datacenter as a computer (Lim et al, 2009), as shown in Fig. 2, leverages the fast progress of the networking capabilities, software defined environments, and the increasing demand on high utilization of computing resources in order to achieve maximal efficiency.

On the networking front, the emerging trend is to utilize a high throughput low latency network as the “backplane” of the system. Such a system can vary from rack, cluster of racks, PoDs, domains, availability zones, regions, and multiple datacenters. During the past 3 decades, the gap between the backplane technologies (as represented by PCIe) and network technologies (as represented by Ethernet) is quickly shrinking. During the next 5 years, the bandwidth gap between PCIe gen 4 (~250 Gb/s) and 100/400 GbE is becoming much less significant.

When the backplane speed is no longer much faster than the network speed, many interesting opportunities arise for refactoring systems and subsystems as these system components are no longer required to be in the same "box" in order to maintain high system throughput. As the network speeds become comparable to the backplane speeds, SSD and storage which are locally connected through a PCIe bus can now be connected through a high speed network. This configuration allows maximal amount of sharing and maximal amount of flexibility to address the complete spectrum of potential workloads. The broad deployment of Software Defined Environments (SDE) within cloud datacenters is facilitating the disaggregation among the management planes, control planes, and data planes within servers, switches and storage (Li et al, 2014).

Systems and subsystems within a composable disaggregated data center are refactored so that these subsystems can use the network "backplane" to communicate with each other as a single system. Composable system concept has already been successfully applied to the network, storage and server areas. In the networking area, physical switches, routing tables, controllers, operating systems, system management, and applications in traditional switching systems are vertically integrated within the same "box". Increasingly, the newer generation switches logically and physically separate the data planes (hardware switches and routing tables) from the control planes (controller and OS and applications) and management planes (system and network management) and allow the disaggregation of switching systems into these three subsystems where the control and management planes can reside anywhere within a data center, while the data planes serve as the traditional role for switching data. Similar to the networking area, storage systems are taking a similar path. Those monolithically integrated storage systems that include HDDs, controllers, caches (including SSDs), special function accelerators for compression and encryption are transitioning into logically and physically distinct data planes - JBOD (just a bunch of drives), control planes (controllers, caches, SSDs) and management planes.

A partially disaggregated memory architecture was proposed by Lim et al (2009, 2012) in which each disaggregated compute node retains a smaller size of memory while the rest of the memory is aggregated and shared remotely. When a compute node requires more memory to perform a task, the hypervisor integrates the local memory and the remote shared memory to form a flat memory address space for the task. During the run time, accesses to remote addresses result in a hypervisor trap and initiate the transfer of the entire page through RDMA (Remote Direct Memory Access) mechanism to the local memory. Their experimental results show an average of ten times performance benefit in a memory-constrained environment. A detailed study of the impacts of network bandwidth and latency of a disaggregated datacenter for executing in memory workloads such as GraphLab, MemcacheD and Pig was reported in Rumble et al. (2011). When the remote memory is configured to contain 75% of the working set, it was found that the application level degradation was minimal (less than 10%) when network bandwidth is 40 Gb/s and the latency is less than $10\mu s$ (Han et al, 2013). Server products based on a disaggregated architecture have already appeared in the marketplace. These include the Cisco UCS M-Series Modular Server, AMD SeaMicro disaggregated architecture, and Intel Rack Scale Architecture as part of the Open Compute Project.

3 SOFTWARE STACK

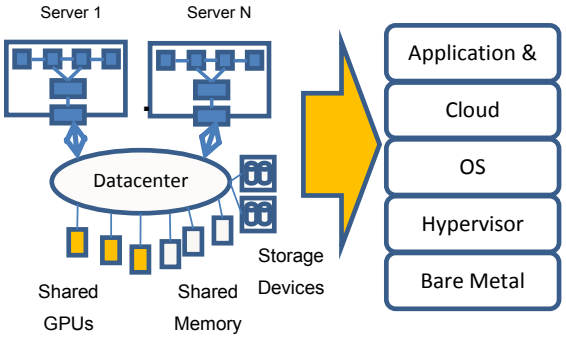

Figure 3: Software stack for accessing disaggregated resources.

Disaggregated datacenter resources can be accessed by application programming models through different means and methods. We consider three fundamental approaches here: (i) hardware based, (ii) hypervisor/operating system based, and (iii) middleware/application based.

The hardware based approach for accessing disaggregated resources is transparent to applications and the OS/hypervisor. Disaggregated memory is mapped to the physical address space of a compute node and is byte addressable across the network. In this case, disaggregated memory is

entirely transparent to the applications. While such transparency is desirable, it forces a tight integration at the memory subsystem either at the physical level or the hypervisor level. At the physical level the memory controller needs to be able to handle remote memory accesses. To avoid the impact of long memory access latencies, we expect that a large cache system is required. Disaggregated GPU and FPGA can be accessed as an I/O device based on direct integration through PCIe over Ethernet. Similar to disaggregated memory, the programming models remain unchanged once the mapping of the disaggregated resource to the I/O address space of the local compute node.

In the second approach, the access of disaggregated resources can be exposed at the hypervisor/container/operating system levels. New hypervisor level primitives - such as getMemory, getGPU, getFPGA, etc. - need to be defined to allow applications to explicitly request the provisioning and management of these resources in a manner similar to *malloc*. It is also possible to modify the paging mechanism within the hypervisor/operatoring systems so that the paging to HDD is now going through a new memory hierarchy including disaggregated memory, SSD, and HDD. In this case, the application does not need to be modified at all. Accessing remote Nvdia GPU through rCUDA (Duato 2010) has been demonstrated, and has been shown to actually outperform locally connected GPU when there is appropriate network connectivity.

Disaggregation details and resource remoteness can also be directly exposed to applications. Disaggregated resources can be exposed via high-level APIs (e.g. put/get for memory). As an example, it is possible to define *GetMemory* in the form of Memory as a Service as one of the Openstack service. The Openstack service sets up a channel between the host and the memory pool service through RDMA. Through *GetMemory* service, the application can now explicitly control which part of its address space is deemed remote and therefore controls or is at least cognizant which memory and application objects will be placed remotely. In the case of GPU as a service, a new service primitive *GetGPU* can be defined to locate an available GPU from a GPU resource pool and host from the host resource pool. The system establishes the channel between the host and the GPU through RDMA/PCIe and exposes the GPU access to applications via a library or a virtual device.

4 NETWORK CONSIDERATIONS

One of the primary challenges for a disaggregated datacenter architecture is the latency incurred by the network when accessing memory, SSD, GPU, and FPGA from remote resource pools. The latency sensitivity depends on how the disaggregated resources are exposed to the programming models in terms of direct hardware, hypervisor, or resource as a service.

The most stringent requirement on the network arises when disaggregated memory is mapped to the address space of the compute node and is accessed through the byte addressable approach. The total access latency across the network cannot be significantly larger than the typical access time of DRAM – which is on the order of 75 ns. As a result, silicon photonics and optical circuit switches (OCS) are likely to be the only options to enable memory disaggregation beyond a rack. Large caches can reduce the impact of remote access. When the block sizes are aligned with the page sizes of the system, the remote memory can be managed as extension of the virtual memory system of the local hosts through the hypervisor and OS management. In this configuration, local DRAM is used as a cache for the remote memory, which is managed in page-size blocks and can be moved via RDMA operations.

Disaggregating GPU and FPGA are much less demanding as each GPU and FPGA are likely to have its local memory, and will often engage in computations that last many microseconds or milliseconds. So the predominant communication mode between a compute node and disaggregated GPU and FPGA resources is likely through bulk data transfer. It has been shown by Reano et al. (2013) that adequate bandwidth such as those offered by RDMA at FDR data rate (56 Gb/s) already demonstrated superior performance than a locally connected GPU.

Current SSD technologies has a spectrum of 100K IOPS (or more) and ~100 us access latency. Consequently, the access latency for non-buffered SSD should be on the order of 10 us. This latency may be achievable using conventional Top-of-the-Rack (TOR) switch technologies if the communication is limited to within a rack. A flat network across a PoD or a datacenter with a two-tier spine-leaf model or a single tier spline model is required in order achieve less than 10 us latency if the communication between the local hosts and the disaggregated SSD resource pools are across a PoD or a datacenter.

5 DISTRIBUTED RESOURCE PROVISIONING

In a disaggregated datacenter with physical resource pooling, it is essential that the physical resources are requested and provisioned with minimum latency so that the use of remote resources will not create a serious performance bottleneck. In this section, we will describe an approach based on distributed scheduling with global shared state in conjunction with predictive resource provisioning.

Resource provisioning and scheduling can be carried out through a centralized, hierarchical, or fully distributed approach. The centralized approach is likely to achieve the optimal resource utilization, but may result in a single point of failure and a severe performance bottleneck. The hierarchical approach, such as the one used in Mesos (Hindman, 2011), allows flexible addition of heterogeneous schedulers for different classes of workloads to a centralized scheduler. The centralized scheduler allocates chunks of resources to the workload specific scheduler, which in turn allocates resources to individual tasks. However, this approach often results in sub-optimal utilization. A fully distributed approach with global shared state, such as the Google Omega (Schwarzkopf, 2013) project, utilizes an optimistic approach for resource scheduling. This approach is likely to perform better as compared to other approaches.

The mechanism for scheduling and provisioning resources from disaggregated physical resource pools starts with the requesting node establishing the type and amount of resource required. As discussed in the previous section, the amount of resource required can be established explicitly by the workload or implicitly as the current requesting node runs out of resource locally. Once the request is received, the resource provisioning engine will identify one or more of the resource pools with available resources, potentially based on the global shared state, for provisioning resources. It will then communicate with the resource manager of the corresponding resource pool to reserve the actual resource. The resource manager for each resource pool commits the resource to the incoming request and resolves the potential conflicts if multiple requests for the same resource occur simultaneously. Once the resource is reserved, the communication between the requesting node and the resource can then commence.

Due to the low latency requirement for provisioning physical resources in a disaggregated datacenter, it is likely that the resources will need to be provisioned and reserved before the actual needs from the workload arise rather than on demand. This may require the resource scheduler to monitor the history of the resource usage so that an accurate workload dependent projection of the resource usage can always be maintained.

6 EXPERIMENTAL RESULTS

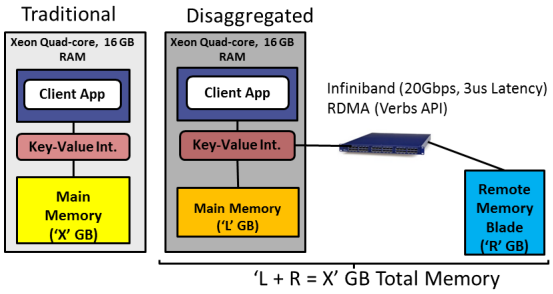

Figure 4: Experimental setup for performance measurement in a disaggregated environment for MemcacheD.

In this section, we describe experiments that demonstrate the workload behavior when a cloud centric application such as MemcacheD is deployed in a disaggregated system environment. In this case, part of client app data is in local DRAM, while the rest is located in the memory of a remote node accessed through an RDMA capable fabric via Verbs API.

The disaggregated infrastructure, as shown in Fig. 4, is entirely transparent to the MemcacheD client. The server side is modified so that the data accessed via key-value interface will be automatically retrieved from either local or remote memory.

The modification is as follows: A small program on another machine allocates a specified amount of memory and registers the allocation with the Infiniband HCA. MemcacheD handshakes with the remote server and obtains the pertinent information such as remote buffer address and access_key. After an initial handshake, it can now perform RDMA reads and writes directly to the remote buffer. The remote buffer is treated as a "victim cache" and is maintained as an append-only log. When MemcacheD runs out of local memory, instead of evicting a key/value pair in the local memory, it now does an RDMA write to the remote memory. When looking up a particular key, it first checks with the local memory (via a hash table). If the key does not exist locally, MemcacheD checks

the remote memory via a locally maintained hash table. If key/value is in the remote memory, it reads in this value through RDMA to a temporary local buffer and sends it to the client. A particular key/value is always either in local memory or remote memory and can never resides in both locations.

The experiments consist of 100,000 operations (95% reads, 5% updates) with uniform random accesses (i.e. no notion of working set as this represents the most challenging situation) running in a single thread.

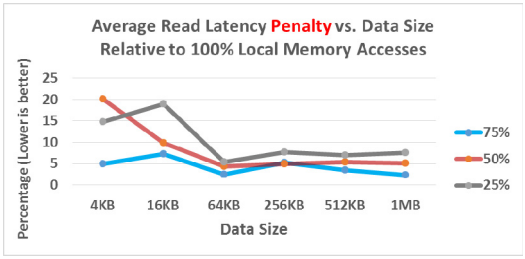

Figure 5: Average read latency penalty vs. data size with respect to 100% local access when the local portion of data varies from 75% to 25%

As shown in Fig. 5, higher percentage of local data always introduces fewer penalties. However, the difference begins to diminish among different ratio of local vs. remote data when the data block size is larger than 64 KB, as larger block size reduces the overhead in the data transfer.

The second set of experiments consist of 100,000 read and update operations (95% reads, 5% updates) with uniform random accesses (i.e. no notion of working set as this represents the most challenging situation) evenly split among 10 threads.

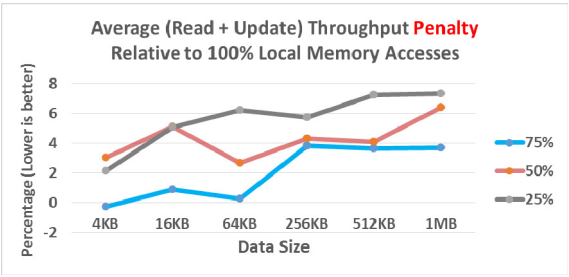

Figure 6: Average read/update throughput penalty vs. data size with respect to 100% local access when the local portion of data varies from 75% to 25%.

As shown in Fig. 6, the throughput penalty is nearly nonexistent when 75% of the access is local and the data size is 4KB. The penalty increases to 2% when only 25% of the access is local. As the data sizes increases, the transfer time of the entire page between the local and the remote node increases, resulting in higher penalty at 4% and 6%, respectively, for 75% and 25% local access.

We can conclude from these experiments that negligible latency and throughput penalty are incurred for the read/update operations if these operations are 75% local and the data size is 64 KB. Smaller data size results in larger latency penalty while larger data size results in larger throughput penalty when the ratio of nonlocal operations is increased to 50% and 75%.

In a second experiment we examine the popular graph analytics platform Giraph, that enables implementation of distributed graph algorithms. In this particular case we populated a 50 node virtual compute cluster with a randomly generated graph of 100 million vertices. The graph is partitioned into 50^2 partitions which are distributed across the compute nodes. We then compute the TopKPagerank properties of the graph. As the computation progresses, messages need to be exchanged to traverse the graph as it crosses node boundaries. Dependent on the connectivity of the graph, the variance in the message creation can result in substantial different memory consumptions per node. Under memory pressure, Giraph will swap entire partitions and messages per vertex to disk using LRU. We examine the memory utilization across the nodes as computations progresses. While cpu utilization is very uniform across all nodes and across the execution of the program, memory utilization varies considerable, which is shown as a heatmap in Fig. 7.

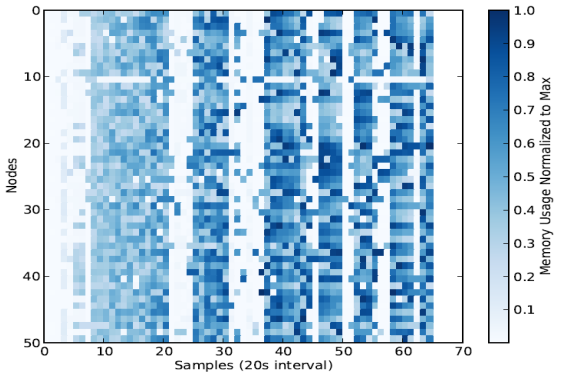

Figure 7: Memory Consumption of Distributed Giraph TopKPagerank application over time.

Analysis of this data reveals that the peak per node memory usage versus the average per node memory has a 2.78:1 ratio, where the aggregate memory usage has a 1.68:1 ratio. We then reduce the per

node memory by a factor of 3 to explore the impact of memory pressure, while the average per node memory is maintained. This increases the overall runtime of the experiment by a factor of 13.8x highlighting that planning resource consumption for best performance requires a memory overprovisioning of a factor of three or alternatively to pay a substantial performance penalty. When the swap disk is on each node is configured to a RamDisk, the overhead reduces to a factor of 6.14x - which is still too high. Having observed the low overheads of RDMA in the MemCacheD example, we stipulate that sharing unused memory across the entire compute cluster instead of through a swap device to a remote memory location can further reduce the overhead. However the rapid allocation and deallocation of remote memory is imperative to be effective.

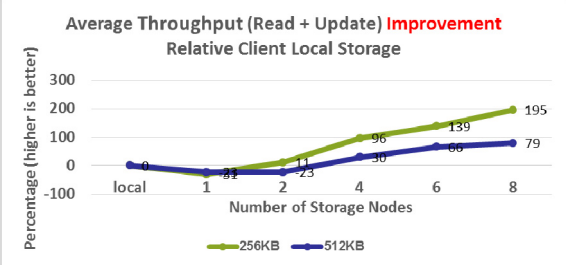

Figure 8: Throughput improvement of disaggregated storage for Cassandra workload.

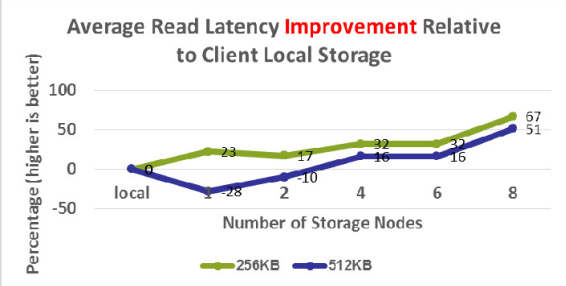

Figure 9: Latency improvement of disaggregated storage for Cassandra workload.

In our final experiment we examine the impact of disaggregated storage. We utilized Cassandra, a popular persistent, i.e disk backed, key value store. In the traditional setup a single server is populated with eight SATA disks that together form the block storage for a ZFS filesystem on which the key value pair storage resides. Ultimately the number of disks in the server is limited to the order of 10s and the SATA v3 bandwidth is limited to 6Gbps. In the disaggregated setup we utilize 4 storage nodes with eight disks each and access to Cassandra was over a 10Gbps Ethernet network. The ZFS cache was limited and data was flushed out of the page cache to ensure that almost all accesses go to disk. A client consisting of 20 threads issued 10K operations (95% read) uniformly accessing the data domain. The bandwidth and latency improvement are shown in Figures 8 and 9. Accessing blocks size of 256KB and 512KB, we observed throughput improvements of up to 195% and 79 % respectively for the disaggregated system case. And latency improvement was 67% and 51%. This experiment substantiates our thesis that accessing data from across multiple disks connected via Ethernet poses less of a bandwidth restriction than SATA and thus improves throughput and latency of data access and obviates the need for data locality. Overall disaggregated storage systems are cheaper to build, manage and incrementally scale and offer higher performance than traditional setups.

As more operations are moved to the shared physical resource pools, it is conceivable that the utilization of shared physical resources will improve, resulting in reduced Capex and/or Opex.

7 CONCLUSIONS

The rapid growth of cloud computing workloads both in terms of the spectrum and volume brought many challenges to the traditional data center design: (1) Fast changing system configuration requirements due to workload constraints; (2) Varying innovation cycles of system components; (3) Maximal sharing of systems and subsystems in order achieve optimal efficiency. The disaggregated architecture provides a promising approach to address these simultaneous challenges. Datacenters based on this architecture allows the refactoring of the datacenter for improved operating efficiency and decoupled innovation cycles among components while the datacenter network becomes the "backplane"of the datacenter.

In this paper, we developed a qualitative assessment of the approaches and challenges for leveraging disaggregated architecture for at scale cloud datacenters. In particular, we compare and contrast the programming models that can be used to access the disaggregated resources, the implications for the network and resource provisioning and management.. Based on this qualitative assessment and early experimental results, we concluded that disaggregated architecture with appropriate programming models and resource provisioning is likely to achieve improved datacentre operating

efficiency for heterogeneous workload environments that can benefit from the improved elasticity of physical resources.

ACKNOWLEDGEMENTS

The authors are grateful for the Dr. Mukil Kesavan for performing the experiments described in Section 6.

REFERENCES

Barroso, Luiz André, Jimmy Clidaras and Urs Hölzle. (2013) The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second edition.

Chang, V. (2014). The business intelligence as a service in the cloud. Future Generation Computer Systems, 37, 512-534.

Duato, J., Pena, A. J., Silla, F., Mayo, R., & Quintana-Ortí, E. S. (2010, June). rCUDA: Reducing the number of GPU-based accelerators in high performance clusters. In High Performance Computing and Simulation (HPCS), 2010 International Conference on (pp. 224-231). IEEE.

Han, S., N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, (2013) Network support for resource disaggregation in next-generation datacenters. In Proc. HotNets.

Herodotou, H., F. Dong and S. Babu. No one (cluster) size fits all: automatic cluster sizing for data-intensive analytics. In Proc. of the 2nd ACM Symposium on Cloud Computing, 2011.

Hindman B., A. Konwinski, M. Zaharia, A. Ghodsi, A. D Joseph, R. H Katz, S. Shenker, I. StoicaMesos (2011): Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center. Proc. ACM USENIX Symposium on Networked Systems Design & Implementation (NSDI), 2011.

Krug, Perry, How Many Nodes? Part 1: An introduction to sizing a Couchbase Server 2.0 cluster. http:// blog.couchbase.com/how-many-nodes-part-1-introduction-sizing-couchbase-server-20-cluster.

Li, C.-S, B. L. Brech, S. Crowder, D. M. Dias, H. Franke, M. Hogstrom, D. Lindquist, G. Pacifici, S. Pappe, B. Rajaraman, J. Rao, R. P. Ratnaparkhi, R. A. Smith and M. D. Williams. (2014) Software defined environments: An introduction. In IBM Journal of Research and Development Vol. 58 No. 2/3 pp. 1-11, March/May.

Lim, K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt and T. F. Wenisch. (2009) Disaggregated Memory for Expansion and Sharing in Blade Servers. In Proc. ISCA.

Lim, K., Y. Turner, J. R. Santos, A. AuYoung, J. Chang, P. Ranganathan and T. F. Wenisch. (2012) System-level implications of disaggregated memory. In Proc. HPCA.

Reano, C., R. May, E. S. Quintana-Orti, F. Silla, J. Duato, A. J. Pena (2013), Influence of InfiniBand FDR on the Performance of Remote GPU Virtualization, IEEE International Conference on Cluster Computing (CLUSTER), pp. 1-8.

Rumble, S. Rumble, D. Ongaro, R. Stutsman, M. Rosenblum, and J. Ousterhout. (2011) It's time for low latency. In Proc. HotOS.

Schwarzkopf, M., Konwinski, A., Abd-El-Malek, M., & Wilkes, J. (2013, April). Omega: flexible, scalable schedulers for large compute clusters. In Proceedings of the 8th ACM European Conference on Computer Systems (pp. 351-364). ACM.

GraphLab. http://graphlab.com/

Memcached - a distributed memory object caching system. http://memcached.org/

PigMix benchmark tool.http://cwiki.apache.org/ confluence/display/PIG/PigMix.

Cisco UCS M-Series Modular Servers. http://www.cisco.com/c/en/us/products/servers-unified-computing/ucs-m-series-modular-servers/index.html.

AMD Disaggregates the Server, Defines New Hyperscale Building Block. http://www.seamicro.com/sites/ default/files/MoorInsights.pdf.

SeaMicro Technology Overview. http://seamicro.com/ sites/default/files/SM_TO01_64_v2.5.pdf.

Intel, Facebook Collaborate on Future Datacenter Rack Technologies, http://newsroom.intel.com/community/ intel_newsroom/blog/2013/01/16/intel-facebook-collaborate-on-future-data-center-rack-technologies, Jan. 2013.

Open Compute Project. http://www.opencompute.org.

SHORT PAPERS

On-demand Text Analytics and Metadata Management with S4

Marin Dimitrov, Alex Simov and Yavor Petkov
Ontotext AD, 47A Tsarigradsko Shose blvd., Sofia, Bulgaria
{marin.dimitrov, alex.simov, yavor.petkov}@ontotext.com

Keywords: Text Analytics, Linked Data, Knowledge Graphs, Semantic Web, Software-as-a-Service, Database-as-a-Service.

Abstract: Semantic technologies provide a new, promising approach for smart data management and analytics. At the same time, the adoption of an emerging technology is usually limited by factors such as its perceived complexity, cost and performance. Startups and mid-size businesses often have very limited resources to evaluate and prototype with emerging technologies, even if their potential for more efficient data management and analytics is significant. The Self-Service Semantic Suite (S4) provides an integrated platform for cloud-based text analytics and Linked Data management as-a-service, so that companies in the early stage of evaluating and adopting semantic technologies can easily access a full suite of semantic data management and text analytics capabilities for smart data analytics in various domains.

1 INTRODUCTION

Some of the typical challenges related to efficient data analytics within enterprises include: valuable information locked in unstructured data sources and available only through inaccurate keyword-based search; a large number of heterogeneous data sources and data silos leading to data quality and reuse problems; slow and rigid data integration processes hindering the access to all the relevant and up-to-date information required for analytics.

Semantic technologies provide a novel approach for data integration, discovery and analytics with significant advantages for a variety of analytical use cases. Among the main advantages of semantic technologies are: the flexible, graph-based RDF data model (W3C, 2014b) which facilitates agile schema-less data integration of heterogeneous data sources; the ability to interlink entities of interest found in text – people, organisations, locations, events and topics – into knowledge graphs; the ability to use formal ontologies as a common metadata layer on top of different data sources and silos; a very expressive RDF query language (SPARQL); the ability to infer implicit facts from data, based on formal reasoning rules; the ability to map and interlink between different schemata and data sources and perform semantic search based on meaning, rather than keywords or schemata; Linked Open Data (Heath, 2011) as a novel data publishing and interlinking paradigm, that can facilitate access to vast amounts of open data on the web.

A number of large companies in various industry sectors – media and publishing, healthcare and life sciences, oil & gas, cultural heritage and digital libraries, retail and finance – have recently adopted semantic technologies in order to achieve higher agility and efficiency when dealing with the modern data analytics challenges. At the same time the wider adoption of semantic technologis is currently limited by factors such as the perceived complexity and cost for deploying semantic data management solutions. Startups and mid-size businesses often have limited resources for evaluating and prototyping with novel data management approaches, while large enterprises often have complex and inefficient procurement processes which hamper the speed of technology adoption and innovation.

2 THE SELF-SERVICE SEMANTIC SUITE

The Self-Service Semantic Suite[1] (S4) aims at reducing the cost and complexity of semantic technology adoption by providing an integrated platform for cloud-based text analytics and Linked Data management as-a-service. With S4 the compa-

[1]http://s4.ontotext.com/

nies in the early stages of evaluating and adopting semantic technologies have the ability to easily and quickly apply a full suite of semantic data management and text analytics capabilities for solutions in various domains, without the need for complex planning, budgeting, provisioning and operations.

The main use cases for text analytics and Linked Data management as-a-service with S4 fall into three broad categories:

- *Reducing Time-to-Market* – companies who are still experimenting with semantic technologies ("technology enthusiasts" and "visionaries", based on their openness towards emerging technology innovations (Moore, 2014)), need capabilities for semantic data management and text analytics that are available from the "get-go" and do not require complex on-boarding, integration and customization, so that the organisations can deliver new products and prototypes at a rapid rate.
- *Reducing Risk* – companies who fall in the group of "pragmatists" when it comes to technology innovation adoption, benefit from a low-cost and low-risk option for experimenting and adopting semantic technologies, without the need to commit to license purchases and hardware provisioning, or deal with inefficient internal procurement processes. By using a platform for semantic data management and text analytics as-a-service, such companies can thoroughly evaluate semantic technologies maturity, reliability, performance and ROI potential before committing to it.
- *Optimising Costs* – even the companies who have already successfully evaluated the potential ROI of semantic technologies and are committed to their long term adoption can often achieve cost reductions by switching to a cloud deployment with pay-per-use cost model.

  S4 provides a self-service and on-demand set of components for semantic data management and text analytics, covering key aspects of the data management lifecycle:
- On-demand and reliable access to central *Linked Open Datasets* such as DBpedia, Freebase, GeoNames, MusicBrainz and WordNet
- A self-managed and a fully-managed scalable *RDF database as-a-service* in the Cloud, for private RDF knowledge graphs
- Various *text analytics services* for news, biomedical documents and social media, which extract valuable insight from unstructured content.

2.1 Linked Data

Linked Data provides a novel data publishing and interchange paradigm that facilitates publishing, interlinking and reuse of large amounts of data within the organisation, or between the organisations within a supply chain, based on four simple principles (Heath, 2011):

- Use URIs as names for things
- Use HTTP URIs, so that people can look up those names
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
- Include links to other URIs, so that they can discover more things.

The amount of openly available Linked Data has been growing at a rapid rate in the recent years (Schmachtenberg, 2014), though various performance and availability problems associated with many public LOD endpoints (Buil-Aranda, 2013) still hamper the wider LOD adoption.

To alleviate the various performance and availability problems associated with open Linked Data, S4 provides a reliable and metered access to key datasets from the LOD cloud via the FactForge[2] large-scale semantic data warehouse (Damova, 2012). More than 5 billion LOD triples, describing 500 million entities, are available to S4 developers via integrated and aligned datasets such as DBpedia, Freebase, GeoNames, and MusicBrainz as well as ontologies and vocabularies like Dublin Core, SKOS and PROTON.

2.2 RDF Databases

RDF databases represent a special class of graph databases where data is modelled based on the semantics of the RDF (W3C, 2014b), and OWL (W3C, 2012) formal model specifications, and data is queried via SPARQL (W3C, 2013). The main advantages of using RDF databases for data management include: the schema agility, ease of integration of heterogeneous data sources, expressive query language, and strong compliance to standards, improved data portability and tool interoperability, as well as ability to infer new, implicit facts from the data.

S4 provides an RDF database-as-a-service capability based on one of the leading enterprise

[2]http://factforge.net/

RDF databases: GraphDB (Bishop, 2011). The cloud database infrastructure of S4 is available in two flavours: a *self-managed* cloud database where the user is in full control of operational aspects – such as availability, performance tuning, backups and restores – and a *fully managed* cloud database where the S4 platform takes care of all aspects related to database administration, provisioning and operations.

The self-managed database in the cloud provides an on-demand and private database server (single tenant model) suitable for organizations that need only the occasional, yet high-performance and reliable access to private RDF datasets, in cases where an on-premise software and hardware deployment would not be cost optimal.

The fully managed RDF database in the cloud provides pay-per-use 24/7 access to private RDF databases and SPARQL endpoints within a multi-tenant model. Operational aspects such as security, availability, monitoring and backups are fully handled by S4 on behalf of the users. The security isolation and resource utilisation control of the different database instances hosted within the same virtual machine in the Cloud is achieved by employing a container-based architecture with the Docker technology.

2.3 Text Analytics Services

Extracting value from text is among the main challenges of Big Data analytics at present, with precious value being locked within vast amounts of unstructured data which is difficult to analyse. Semantic technologies are a good fit for dealing with the "variety" aspect of Big Data, and structured, semi-structured and unstructured data sources can be interlinked into semantic (RDF) graphs.

S4 provides various services for real-time text analytics:

- *News Analytics* – the service performs information extraction and entity linking to large open knowledge graphs such as DBpedia, Freebase and GeoNames (Damova, 2012). The text analysis process is a combination of rule-based and machine learning techniques (Georgiev, 2013).
- *News Classifier* – the service performs categorisation of news articles according to the 17 top-level categories of the IPTC Subject Reference System (IPTC, 2003).
- *Biomedical Analytics* – the service can recognize more than 130 biomedical entity types (Georgiev, 2011) and semantically link them to a large-scale biomedical LOD knowledge base (LinkedLifeData[3]).
- *Twitter Analytics* – the service is based on the TwitIE open source microblog analysis pipeline (Bontcheva, 2013) and it performs named entity recognition of various classes of entities as well as normalisation of most common abbreviations frequently found in tweets.

3 ARCHITECTURE

The architecture of S4 is based on best practices and design patterns for scalable AWS cloud architectures (AWS, 2014).

3.1 Public Cloud Platform

S4 is currently deployed on a public AWS[4] cloud platform and it utilizes various cloud infrastructure services such as:

- *distributed storage* via Simple Storage Service (S3), Elastic Block Storage (EBS), and DynamoDB
- *elastic computing* via Elastic Compute Cloud, Auto Scaling and Elastic Load Balancer
- *application integration* via the Simple Queue Service (SQS) and Simple Notification Service (SNS)

3.2 S4 Architecture

S4 follows the principles of micro-service architectures and it is comprised of the following main components and layers (Figure 1):

- *Load Balancer* – the entry point to all S4 services is the AWS load balancer which redirects incoming requests to one of the available routing nodes.
- *Routing Nodes* – these instances host various micro-service frontends to the text analytics nodes, the LOD as well as the database nodes. The job of these nodes is just to perform pre-processing and post-processing (if necessary) and to forward the client request (a document for text processing or a database query/update) to the proper backend node – a text analytics node, a database node, or the LOD server itself. All instances host the same set of stateless front-end services and this layer is automatically scaled up or down (new instances added or removed) based

[3]http://linkedlifedata.com/
[4]http://aws.amazon.com/

on the current system load and performance. The communication between the routing nodes and the LOD server and database nodes is synchronous, while the communication with the text analytics nodes is asynchronous (via a distributed queue).

- *Text Analytics Nodes* – these instances are responsible for processing the text documents sent for analysis to S4. They host the different text analytics services for news, biomedical documents and social media. This layer is also automatically scaled up or down based on the current system load and performance.
- *Database Nodes* – a database node is a virtual machine that hosts a number of independent GraphDB instances packaged as Docker containers. Each database container stores its data on a dedicated network-attached storage volume (EBS), and EBS volumes are not shared between different database containers for improved performance and isolation. New database nodes are dynamically added by the *Coordinator* node when there are no free database container slots left on the current database nodes. If a database node has free container slots then it periodically contacts the coordinator for the IDs of databases which are still waiting to be deployed, so that the database node can fill up its full hosting capacity as soon as possible – by attaching the dedicated EBS volume for the database to one of its available containers.
- *Coordinator* – there is a single coordinator which is responsible for distributing the database initialisation tasks among the active database nodes. The coordinator keeps a "routing table" with information on the databases hosted by each database node. If a database node crashes, then the coordinator will mark its databases as non-operational and will re-distribute them (their IDs) to other database nodes with free database slots, or to the new database node which will be automatically instantiated by the AWS Auto Scaler to replace the crashed node.
- *Linked Data Server* – currently the LOD data available through S4 is hosted on the FactForge semantic data warehouse.
- *Integration Services* – a distributed queue and a distributed push messaging service are used for the loose coupling and asynchronous communication between the components of the platform. For example, the requests for text processing are first handled by a routing node, which puts a processing request in the distributed queue (SQS) so that one of the available text analytics nodes will pull the request, process it, and send the result back to the routing node. This way the routing and text analytics nodes are not aware of their number and topology and they can be scaled up/down independently. In a similar manner, the distributed push messaging service (SNS) is used for loose coupling between the database nodes, routing nodes and the coordinator. Each database node sends "heartbeats" several times per minute via the notification service, so that routing nodes and the coordinator get a confirmation that the databases hosted by that node are still operational. Each database node also sends periodic updates regarding the databases it is hosting, so that the routing nodes and the coordinator can update their routing tables if necessary.
- *Distributed Storage* – AWS Simple Storage Service (S3) is used for transient storage of documents, and for database backups. The data for the database nodes is stored on high-performance network-attached storage volumes (EBS).
- *Metadata Store* – a distributed key-value store (DynamoDB) stores simple metadata regarding the databases hosted on the platform (user ID, dedicated EBS volume ID, configuration parameters, text analytics components metadata, user accounts, etc.), as well as the logging data from all platform operations.
- *Management and Monitoring Services* – various management microservices cover operational aspects such as logging, reporting, account management, quota management and operations monitoring.

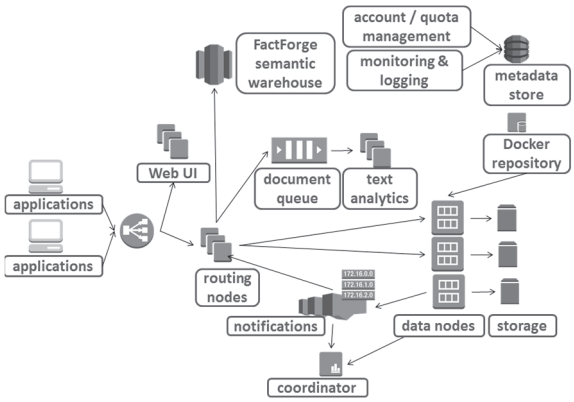

Figure 1: S4 architecture.

3.3 Operations

The behaviour of each of the S4 sub-systems (text analytics, Linked Data server, database as-a-service) is described next.

3.3.1 LOD Access

Since S4 just provides metered access to the FactForge semantic warehouse (Damova, 2012), the job of the routing node that receives a request for the LOD server (a SPARQL query for some LOD dataset) is to just forward the query to FactForge and then return the result back to the client application in a synchronous manner. There is only one instance of the FactForge warehouse.

3.3.2 Text Analytics

When a routing node receives a document to be processed by one of the text analytics services on S4, it packages the document with additional metadata and puts the request in the distributed queue (SQS). One or more text analytics nodes are constantly poling the queue for pending requests for document processing. When a text analytics node retrieves a request from the queue, it will process it and return the result directly to the routing node that originated the request (this information is available in the metadata of the request), so that the routing node can in turn return the result to the client application that sent the document for processing. Neither the routing nodes, nor the text analytics nodes are aware of the exact number of topology of the nodes and new nodes can be added or removed dynamically as needed.

3.3.3 Database As-a-Service

Unlike the text analytics sub-system, the routing nodes need to be aware of the exact topology of the database nodes, since a client request has to be directed to the proper database node hosting the client database at that particular moment. For this reason, the database as-a-service sub-system is more complex and it involves a Coordinator node and a distributed push messaging service for asynchronous communication between the routing nodes, the database nodes and the Coordinator.

Each routing node maintains a routing table so that it knows which database node is currently hosting a particular client database: the routing data for each database includes the IP address (of the database node) and the port (of the Docker container running on the database node) where the database is currently hosted. If there is a change in the database layer topology (e.g. a database node crashes and another one is instantiated to replace it and host its databases) then the routing tables are updated.

When a routing node starts, it immediately subscribes to the distributed notifications service (SNS), so that it can start receiving heartbeats and routing updates from the database nodes. If a client request is forwarded by the load balancer to the new routing node before it has its routing table fully initialised, then the node will just queue the client request locally. After a short period of time the routing node will receive the heartbeat and routing notifications from all database nodes, so that it can start forwarding client requests to the proper database node. After a routing node forwards the client request to the proper database node, it waits for the database response and then forwards the response back to the originating client application (the communication between the client application the routing node and the database node is synchronous).

When the Coordinator starts, it first reads the metadata about all databases on the platform from the metadata store, and then subscribes to the notifications service (SNS) in order to receive heartbeats and routing updates from the database nodes. If after short period of time there is still a database (listed in the metadata store) that no database node is currently hosting, the coordinator assumes that this database is down and will send its ID to the next database node which contacts the coordinator requesting mew databases for initialisation.

When a database node starts, it initialises its database containers from the local Docker repository and immediately contacts the coordinator to request a list of databases (IDs) which to host on its local containers. When the coordinator provides the information, the database node just attaches the dedicated network-attached storage (EBS) volume for the database to itself and performs the final OS level configuration so that the database container can be fully initialised. At this point, a Docker container with a running GraphDB instance and an attached data volume is fully operational on the database node. The next step for the database mode is to subscribe to the notifications service and start sending regular heartbeats and routing updates to the routing nodes and the coordinator. At this point the database node is ready to serve client requests forwarded to its active databases by the routing nodes.

3.4 Dealing with Failure

3.4.1 LOD Access

The FactForge semantic warehouse is single-point of-failure component related to LOD access, since due to its large scale and hardware requirements it is not cost efficient at present to provide multiple replicas of the warehouse for improved availability. Nonetheless, the warehouse availability and performance is constantly being monitored, and FactForge is listed as the LOD endpoint with the highest availability and reliability in a recent analysis by (Buil-Aranda, 2013).

3.4.2 Text Analytics

In the case of a routing node failure the load balancer will automatically stop redirecting requests to the problematic node and the Auto Scaler will instantiate a new replacement node. Only the currently open connections from client applications to the problematic routing node will be terminated abnormally, while the rest of the system will be fully operational. If a text analytics node fails while processing documents, then after a short period of time the documents that it was processing will become visible as messages in the distributed message queue again, so that a different healthy backend node can pull them for processing (messages are deleted from the queue only upon returning the result to the client application, and marked as "invisible" while being processed by some text analytics node).

3.4.3 Database As-a-Service

In the case of a routing node failure the load balancer will start redirecting requests to other (healthy) routing nodes. Meanwhile the AWS Auto Scaler will instantiate a new routing node to replace the failed one. The new node follows the initialisation steps described in the previous section and it will be soon fully operational. Database nodes and the Coordinator are not affected by a routing node failure.

If the Coordinator crashes, the Auto Scaler will automatically instantiate a replacement coordinator node. The new coordinator will follow the standard initialisation steps, build its routing table and start distributing non-operational database IDs when requested by the database nodes with free containers.

If a database node crashes it will stop sending heartbeats to the notification service and the routing nodes will be aware that the databases hosted on that database node are not operational, so that they will start queueing the client requests for these databases locally. The Coordinator will mark the databases hosted by the failed data node as non-operational and will be ready to send their IDs to the next database node which requests pending databases for initialisation. Meanwhile, the Auto Scaler will detect the node failure and instantiate a replacement node, which will follow the initialisation steps from the previous section: first request pending databases from the coordinator, then start sending heartbeats and routing updates to the routing nodes and the coordinator. Soon the routing nodes will be aware of the new location of the failed databases and will start forwarding the client requests that were being queued.

A combination of nodes may crash at any time (including *all* of the nodes on the platform) but the recovery will always follow the steps above, with the following dependencies:

- The coordinator does not depend on any active routing / database nodes to be operational.
- Database nodes depend on the coordinator to be operational, so that they can receive database initialisation tasks for their hosted containers.
- Routing nodes depend on the database nodes to be operational, so that they can initialise their routing tables and forward client requests to the proper database

3.5 Scalability

Routing nodes are dynamically added based on the current system load. When a new routing node is added, the load balancer will automatically start redirecting some of the incoming requests to it.

Text analytics nodes are also dynamically added when the number of documents waiting for processing in the queue exceeds a pre-defined threshold.

The integration services (distributed message queue and distributed push messaging) are designed by AWS so that they can scale up to thousands of messages processed per second.

The database nodes are currently not replicated, so if a particular database experiences a usage spike and becomes overloaded, its performance will temporarily decrease. At the same time, due to the simple container and network-attached storage based architecture that S4 employs, it is possible to quickly scale up the database container by re-deploying it on a bigger virtual machine with more memory and CPU cores. EBS volume performance can also be

increased (at a higher cost).

4 CONCLUSIONS AND FUTURE WORK

This paper presents the Self-Service Semantic Suite (S4), a cloud platform providing on-demand access to key capabilities for semantic data management: access to large open knowledge graphs (Linked Open Data), RDF databases as-a-service, and various text analytics services.

Several existing platforms offer text analytics as-a-service capabilities: *Alchemy*, *Bitext*, *DatumBox*, *MeaningCloud*, *OpenCalais*, *OpenAmplify*, *Saplo*, *Semantria*, etc. *Dydra* provides capabilities for RDF databases as-a-service. Some of the platforms for text analytics as-a-service already cover multiple languages and provide support for sentiment analytics as well. The main differentiation of the S4 platform is that it provides an integrated suite for semantic analytics which allows for content from unstructured data sources to be semantically enriched and interlinked into an RDF knowledge graph, so that semantic search and discovery can be utilised for data analytics.

S4 has already been deployed in production[5], but a variety of improvements are planned or already in development:

- Availability of multilingual text analytics services, including services for sentiment analytics;
- Asynchronous, batch processing of large volumes of text, in addition to the current synchronous mode of operation;
- Adoption of JSON-LD (W3C, 2014a) for the text analytics services output;
- Integration of 3rd party tools for visual exploration and navigation of large scale Linked and RDF data;
- Integration of Linked Data Fragment based containers (Verborgh, 2014), as an alternative approach for scalable querying of RDF data.

ACKNOWLEDGEMENTS

Some of the work related to S4 is partially funded by the European Commission under the 7th Framework Programme, project DaPaaS[6] (No. 610988).

[5]http://s4.ontotext.com/
[6]http://project.dapaas.eu/

REFERENCES

Amazon Web Services, 2014. AWS Reference Architectures. Available at http://aws.amazon.com/architecture

Bishop, B., Kiryakov, A., Ognyanoff, D., Peikov, I., Tashev, Z., Velkov, R., 2011. OWLIM: A family of scalable semantic repositories. In *Semantic Web Journal, vol 2, number 1*.

Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M.A., Maynard, D., Aswani, N., 2013. TwitIE: An Open-Source Information Extraction Pipeline for Microblog Text. In *RANLP'2013*, *International Conference on Recent Advances in Natural Language Processing.*

Buil-Aranda, C., Hogan, A., Umbrich, J., Vandenbussche, P., 2013. SPARQL Web-Querying Infrastructure: Ready for Action? In *ISWC'2013, 12th International Semantic Web Conference.*

Damova, M., Simov, K., Tashev, Z., Kiryakov, A., 2012. FactForge: Data Service or Diversity through Inferred Knowledge over LOD. In *AIMSA'2012, 15th International Conference on Artificial Intelligence: Methodology, Systems and Applications.*

Georgiev, G., Pentchev, K., Avramov, A., Primov, T., Momtchev, V., 2011. Scalable Interlinking of Bio-Medical Entities and Scientific Literature in Linked Life Data. In *CALBC'2011*, *Workshop on Collaborative Annotation of a Large Biomedical Corpus*.

Georgiev, G., Popov, B., Osenova, P., Dimitrov, M., 2013. Adaptive Semantic Publishing. In *WaSABi'2013*, *Workshop on Semantic Web Enterprise Adoption and Best Practices.*

Heath, T., Bizer, C., 2011. *Linked Data: Evolving the Web into a Global Data Space.* Morgan & Claypool.

IPTC, 2003. Subject Reference System Guidelines. Available at http://www.iptc.org/std/NewsCodes/0.0/documentation/SRS-doc-Guidelines_3.pdf

Schmachtenberg, M., Bizer, C., Paulheim, H., 2014. Adoption of Linked Data Best Practices in Different Topical Domains. In *ISWC'2014, 13th International Semantic Web Conference.*

Moore, G., 2014. *Crossing the Chasm: Marketing and Selling Disruptive Products to Mainstream Customers*, Harper Business, 3rd edition.

Verborgh, R., Hartig, O., De Meester, B., Haesendonck, G., De Vocht, L., Vander Sande, M., Cyganiak, R., Colpaert, P., Mannens, E., Van de Walle, R., 2104. Querying Datasets on the Web with High Availability. In *ISWC'2014, 13th International Semantic Web Conference.*

W3C, 2012. OWL 2 Web Ontology Language Document Overview (2nd Edition). W3C Recommendation (December 2012). Querying Datasets on the Web with High Availability. In *ISWC'2014, 13th International Semantic Web Conference.*

W3C, 2013. SPARQL 1.1 Overview. W3C Recommendation (March 2013).

W3C, 2014a. JSON-LD 1.0 – a JSON-based Serialisation for Linked Data. W3C Recommendation (January 2014).
W3C, 2014b. RDF 1.1 Concepts and Abstract Syntax. W3C Recommendation (February 2014).

Evaluation Metrics for VM Allocation Mechanisms in Desktop Clouds

Abdulelah Alwabel, Robert Walters and Gary Wills
School of Electronics and Computer Science, University of Southampton, Southampton, U.K.
{aa1a10, rjw1, gbw}@ecs.soton.ac.uk

Keywords: Cloud Computing, Desktop Clouds, Evaluation Metrics, Node Failures, Throughput, Availability, Power Consumption, DesktopCloudSim.

Abstract: Desktop Cloud computing is the idea of benefiting from computing resources around us to build a Cloud system in order to have better usage of these resources instead of them being idle. However, such resources are prone to failure at any given time without prior knowledge. Such failure events have a can negative impact on the outcome of a Desktop Cloud system. This paper proposes metrics that can evaluate the behaviour of Virtual Machine (VM) allocation mechanisms in the presence of node failures. The metrics are throughput, power consumption and availability. Three VM allocation mechanisms (Greedy, FCFS and RoundRobin mechanisms) are evaluated using the given metrics.

1 INTRODUCTION

Desktop Cloud computing is the idea of benefiting from computing resources around us to build a Cloud system in order to have better usage of these resources instead of them being idle (Alwabel et al., 2014a). Desktop Cloud computing is an alternative to the traditional way of providing Cloud services. Traditionally, Cloud service providers, such as Amazon, dedicate a massive number of computer nodes that are located in one or more data centres to provide services over the Internet (Buyya et al., 2009). The idea of Desktop Cloud is stimulated by the success of Desktop Grid to offer Grid services using resources contributed by people over the Internet (Anderson et al., 2002).

There are several research issues in Desktop Clouds that need further attention from researchers. Research issues are security and privacy; resource management; and node failures (Alwabel et al., 2014a). Node failure rates in Desktop Cloud are reported to be quite high and can affect the performance of Desktop Clouds (Alwabel et al., 2014b). It is proposed that a Virtual Machine (VM) allocation mechanism can play an important role in order to reduce the negative effect of node failures (Alwabel et al., 2015a). This paper proposes metrics that can be used to evaluate the behaviour of a VM allocation mechanism. Section 2 of this paper gives an overview of Desktop Cloud. Next section proposes and discusses the evaluation metrics. The third section presents our findings of employing the metrics to evaluate several VM allocation mechanisms from the literature. A conclusion and future is presented in the last section.

2 DESKTOP CLOUD COMPUTING

Desktop Cloud computing is a new type of Cloud built using resources that would otherwise remain idle and unused (Alwabel et al., 2014a). For example, most PCs in universities remain idle and unused after 5 pm. The idea of Desktop Cloud is motivated by the success of Desktop Grids (Kondo et al., 2004). The concept of Desktop Grid is to exploit normal computing resources such as PCs and laptops to process and execute Grid tasks. Several Desktop Grid projects have proven success in achieving this goal such as SETI@home (Anderson et al., 2002).Desktop Cloud merges two ideas: Desktop Grids and Cloud computing. Note that“Desktop” term is derived from Desktop Grids because both of Desktop Clouds and Desktop Grids are mainly based on desktop PCs and laptops. whilethe term “Cloud” comes from Cloud since Desktop Cloud provides services based on the Cloud

business model. Several synonyms are used which mean Desktop Cloud, such as Ad-hoc Cloud, Volunteer Clouds and Non-Dedicated Clouds. The literature shows that very little work has been carriedout in this research area.

"Ad-hoc Cloud"(Kirby et al., 2010) is the idea of employing distributed resources within an organisation to form a Cloud. "Nebula" Chandra and Weissman, 2009; Weissman et al., 2011) is a research project that aims to use distributed resources with an aimof creating a volunteer Cloud which offers services free of charge. "Cloud@home"(Cunsolo and Distefano, 2010; Cunsolo et al., 2009) is a project implementing the "@home" philosophy in Cloud computing. The goal of Cloud@home is to establish a new model of Cloud computing built on resources that are donated by individual users over the Internet. Further to that, CERN has recently announced an initiative to bring their Desktop Grid project, which is called LHC@home, into the Cloud (Harutyunyan et al., 2012). It is suggested that non-dedicated resources can be used by Cloud providers when their local infrastructure cannot meet demands ofCloud consumers at peak times (Andrzejak et al., 2010).

Desktop Clouds can be formed into private Clouds or public Clouds. The first scenario to build a private Desktop Cloud can be considered as follows:supposea university wishes to benefit from its computing resources to form a Cloud. The resources can be of any type ranging from PCs to servers etc, each computing resource is called a Cloud node when it joins the Cloud. Researchers and staff within the universitycan benefit from this Desktop Cloud by submitting their requests to acquire Cloud services. Requests are processed in the virtualisation layer on top of Cloud physical nodes. Another scenario that can be considered is a public Desktop Cloud that allows people to contribute their own computing resources to be used by Cloud clients (Cunsolo et al., 2009).The people are invited to contribute their machines when these resources become idle in order to form a Desktop Cloud. People can be motivated to participate by telling them that such projects can serve science and research communities. Another incentive might be being permitted to use the Desktop Cloud resources when they want them.

One of the main issues in Desktop Clouds is the high rate of node failures during run time (Alwabel et al., 2014b). In Desktop Cloud computing, node failure events can include any event that causes the node to leave the Cloud for any reason. Next section proposes several metrics that can be used to evaluate the outcome of a VM allocation mechanism in the presence of node failures.

3 EVALUATION METRICS

The efficiency of Cloud computing is defined by a set of evaluation metrics. Employing efficient metrics for Cloud computing is vital in order to optimise the Clouds. It has been shown that there is no systematicanalysis for evaluation metrics for Cloud Computing (Li et al., 2012).The diversity of architectures of Cloud providers requires evaluation metrics to be platform independent(Goiri et al., 2012). However, the literature shows there are several studies assessing the service provided by the Cloud from the prospective of customers. Most of the literature (such as (Lenk et al., 2011), (Stantchev, 2009) and (Villegas et al., 2012)) focuses on the cost-performance of services in order to adopt a better decision-making policy that can help customers to choose a service provider according to their requirements. For example, some customers can tolerate some performance degradation in exchange for low cost of service.

A Virtual Machine (VM) allocation mechanism can play an important part in the outcome of a Cloud system. In this work, we considered three metrics that can be used to evaluate a VM allocation mechanism implemented in a Desktop Cloud. VM allocation mechanism is the process of allocating a VM to a Physical Machine (PM) (Alwabel et al., 2014b). The metrics are throughput, power consumption and availability. They are discussed further in the following subsections.

3.1 Throughput

Throughput is an important metric to measure the outcome of a Cloud system in the presence of node failures. Throughput metriccalculates the number of successfully completed tasks *st* that are submitted by clients out of the total number of submitted tasks *tt* (Garg et al., 2013). Throughput is calculated as follows:

$$throughput = 100 * \frac{\sum st}{tt}$$

Most papers in the literature focus on the performance notion which includes attributes such as response time and average turnover time such as (Van et al., 2010) and (Stantchev, 2009). This is because researchers assume that Cloud nodes are very reliable (Buyya et al., 2010). However, we

consider throughput because it is known that node failures in Desktop Clouds are norms rather than exceptions (Abdulelah Alwabel et al., 2014b).

3.2 Power Consumption

Power consumption metric considers the amount of energy *pwr* that is consumed by each node in the infrastructure layer of a Cloud system. It is measured by Kilo Watt hour (kWh). The metric of power consumption is given as follows:

$$power\ consumption = \sum_{i=0}^{n} pwr(node_i)$$

Beloglazov et al., (2012) set power consumption as one of the metrics to measure the outcome of their energy-aware resource allocation algorithm for Cloud computing. Energy efficiency can be defined as the number of instructions in billions executed per Watt hour (Bash et al., 2011). The Standard and Performance Evaluation Corporation (SPEC) community released SPECpower metric to measure power consumption (Lange, 2009). SPECpower is a Java application that generates a set of transactions completed per second. SPECpower calculates energy consumed by total number of operations in Watt-hours. Energy consumption is considered a metric for evaluating the proposed model in Desktop Clouds.

3.3 Availability

Availability means how much computing power is available to accommodate new VM requests. The failure of nodes can affect the availability of Desktop Clouds. A question in this context is whether the employed VM allocation mechanism can help in improving node availability. Let *avl* denote the availability of a Cloud node while the total computing power of all Cloud nodes is denoted*tot.cp*. The availability is given as follows:

$$availability = \frac{\sum available\ nodes}{tot.cp}$$

4 EXPERIMENT

The experiment is conducted to evaluate three VM mechanisms which are First Come First Serve(FCFS) (Schwiegelshohn and Yahyapour, 1998), Greedy (Cunha et al., 2001) and RoundRobin (Rasmussen and Trick, 2008).These mechanisms are evaluated using the metrics proposed in the previous section.

4.1 Experiment Design

A Desktop Cloud was simulated using DesktopCloudSim (Alwabel et al., 2015b) simulation extension to CloudSim (Calheiros et al., 2011). CloudSim is a widely used simulation tool to simulate the behaviour of a Cloud System. DesktopCloudSim enables researchers to simulate failure events happening within the infrastructure level of a Cloud (i.e., enabling Cloud nodes to fail during run time). In order to simulate a Desktop Cloud, data of a Desktop Grid system retrieved from Failure Trace Archive was used to simulate both the infrastructure of a Desktop Cloud since both Desktop Cloud and Desktop Grid use infrastructure similar to each other (Alwabel et al., 2015a). Secondly, the archive provides name of the machine that fails along with the time of failure. Another input to the simulation tool is the workload containing tasks submitted to be executed. The workload is collected from PlanetLab archive (Peterson et al., 2006).

The Experiment assumes that 700 instances of VMs are requested to run for 24 hours. The types of VM instances are: *micro, small, medium and large*. The VM instances are similar to VM types that are offered by Amazon EC2. The type of each given VM instance is randomly selected. The number of VM instances and types remain the same for all run experiment sets. Each VM instance processes a bunch of tasks fromthe given workload.

It is assumed in the experiment that if a node fails then all VMs on this node will be lost. Destroying a VM instance causes all running tasks on the VM to be destroyed which consequently affectsthe throughput (i.e., these tasks are considered failed tasks). The destroyed VM will berestarted on another PM and begin to receive new tasks. Any failed node which recovers may rejoin the Cloud. The experiment is run 180 times, each time is a run for one day in the simulation. 180 days represents six-month period.The experiment was simulated and run on a Mac i27 (CPU = 2.7 GHz Intel Core i5, 8 GB MHz DDR3) with operating system OS X 10.9.4. The results were processed and analysed using IBM SPSS Statistics v21 software.

Table 1: Throughput Metric.

Mechanism	Mean (%)	Median (%)	Variance	Standard Dev.
FCFS	79.21	78.77	37.03	6.09
Greedy	88.61	89.48	16.85	4.1
RoundRobin	85.47	85.29	15.13	3.89

4.2 Results and Discussion

Table 1 shows a summary of results obtained when measuring the throughput metric for each VM allocation mechanism in the experiment. Kolmogorov-Smirnov (K-S) test (Field, 2009) of normality shows that the normality assumption was not satisfied because the FCFS and Greedy mechanisms are significantly non-normal, $P < .05$. Therefore, the non-parametric test Friedman's ANOVA (Field, 2009) was used to test which mechanism can yield better throughput. Friedman's ANOVA test confirms that throughput varies significantly from mechanism to another, $X_F^2(2) = 397.14, P < .001$. Mean, median, variance and standard deviations are report in Table 1.

Three Wilcoxon pairwise comparison tests (Field, 2009) were used to find out which mechanism gave the highest throughput. Note that three tests are required to compare threepairs of mechanisms which are FCFS vs. Greedy, FCFS vs. RoundRobin and Greedy vs. RoundRobin mechanisms. The level of significance was set to 0.017 using Bonferroni correction (Field, 2009) method because there were three post-hoc tests required (.05/3 ≈ .017). The tests show that there is a statistically significant difference between each mechanism with its counterparts. Therefore, we can conclude that Greedy mechanism produces highest throughput since it has the median with highest value (median = 89.48%).

Table 2 reports the mean, median, variance, standard deviation when power consumption was measured in the experiment. Friedman's ANOVA test was applied to the power consumption results to show if that there a significant difference between the mechanisms, $X_F^2(2) = 540, P < .001$.Friedman's ANOVA test was selected because the power consumption results are not all distributed normally since the critical value (p-value) <0.5 for FCFS and Greedy mechanisms results.

Table 2: Power Consumption Metric.

Mechanism	Mean (kWh)	Median (kWh)	Variance	Standard Deviation
FCFS	533	538	867	29.45
Greedy	638	641	738	27.16
RoundRobin	1884	1883	22237	149

Three Wilcoxon tests were conducted to identify which mechanism consumes the least power. The tests showed that there is a statistically significant difference between each pair of mechanisms. Therefore, the FCFS mechanism consumes significantly less power among the testes for mechanism because the median of power consumption of the FCFS is 538 kWh.

Table 3 shows a summary of descriptive results obtained when measuring the availability metric for each VM allocation. Since the results are not normally distributed, Friedman's ANOVA test was used to test which mechanism can yield better availability. Friedman's ANOVA test confirms that availability varies significantly from mechanism to another, $X_F^2(2) = 510.78, P < 0.001$. Mean, median, variance and standard deviations are reported in Table 3.

Three Wilcoxon pairwise comparison tests were used to find out which mechanism produced best availability. The tests show that there is a significant difference between each pair of VM mechanisms. Greedy mechanism outperformed other mechanisms in terms of availability by looking at the median (86.23%).

The results show that the throughput, power consumption and resource availability can be affected by node failures and thus, yield different outcomes according to the implemented mechanism. According to this experiment, Greedy mechanism yields the best throughput and availability while the FCFS mechanism consumesleast power. A note worth mentioning from our experiment is that at least10% of submitted tasks failed because of node failures. Therefore, there is actual need to implement a fault-tolerant mechanism for Desktop Cloud.

Table 3: Availability Metric.

Mechanism	Mean (%)	Median (%)	Variance	Standard Deviation
FCFS	85.03	84.59	4.21	2.05
Greedy	86.22	86.23	3.09	1.76
RoundRobin	81.98	81.91	2.44	1.6

5 CONCLUSIONS AND FUTURE WORK

Desktop Cloud computing is a new type of Cloud computingwhich aims to employ computing resources to build a Cloud system. The resources that are employed in Desktop Clouds are normal computing resources such PCs and laptops. These resources would remain idle and unused if they are not used within a Desktop Cloud system.The model of Desktop Cloud is to move Desktop Grid systems towards Cloud computing era. This paper presented throughput, power consumption and availability as metrics that can be used to evaluate VM allocation mechanisms.

The FCFS, Greedy and RoundRobin VM allocation mechanisms were evaluated using the proposed metrics. The experiment was conducted using DesktopCloudSim simulation tool which enables researchers to simulate Desktop Cloud systems. Our findings showed that Greedy mechanism can give better in terms of throughput and availability while the FCFS mechanism can consume the least power among other mechanisms.

Our findings showed that the failure of tasks can reach up to 10% of all submitted tasks as a result of node failures. Therefore, our future work is to develop a new fault-tolerant VM mechanism for a Desktop Cloud system. In addition to that, researchers should pay attention to power consumed by Cloud nodes in order to reduce it. The reduction of power consumption can result in reducing the running costs of Desktop Clouds.

REFERENCES

Alwabel, A., Walters, R., Wills, G.B., 2014a. A view at desktop clouds. In: ESaaSA 2014.

Alwabel, A., Walters, R., Wills, G.B., 2014b. Evaluation of Node Failures in Cloud Computing Using Empirical Data. Open J. Cloud Comput. 1, 15 – 24.

Alwabel, A., Walters, R., Wills, G.B., 2015a. A Resource Allocation Model for Desktop Clouds. In: Delivery and Adoption of Cloud Computing Services in Contemporary Organizations.

Alwabel, A., Walters, R., Wills, G.B., 2015b. DesktopCloudSim : Simulation of Node Failures in The Cloud. In: The Sixth International Conference on Cloud Computing, GRIDs, and Virtualization CLOUD COMPUTING 2015. iaria, Nice.

Anderson, D., Cobb, J., Korpela, E., Werthimer, D., Anderson, P., Lebofsky, M., 2002. SETI@home An Experiment in Public-Resource Computing. Commun. 45.

Andrzejak, A., Kondo, D., Anderson, D.P., 2010. Exploiting non-dedicated resources for cloud computing. 2010 IEEE Netw. Oper. Manag. Symp. - NOMS 2010 341–348.

Bash, C., Cader, T., Chen, Y., Gmach, D., Kaufman, R., Milojicic, D., Shah, A., Sharma, P., 2011. Cloud Sustainability Dashboard, Dynamically Assessing Sustainability of Data Centers and Clouds. In: Proceedings of the Fifth Open Cirrus Summit. Moscow.

Beloglazov, A., Abawajy, J., Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. Futur. Gener. Comput. Syst. 28, 755–768.

Buyya, R., Broberg, J., Goscinski, A., 2010. Cloud Computing Principles and Paradigms. John Wiley & Sons.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I., 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Futur. Gener. Comput. Syst. 25, 599–616.

Calheiros, R., Ranjan, R., Beloglazov, A., De Rose, C´.A.F., Buyya, R., 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. … 23–50.

Chandra, A., Weissman, J., 2009. Nebulas: Using distributed voluntary resources to build clouds. In: Proceedings of the 2009 Conference on Hot Topics in Cloud Computing. USENIX Association, pp. 2–2.

Cunha, J., Kacsuk, P., Winter, S., 2001. Parallel Program Development for Cluster Computing: Methodology, Tools and Integrated Environments. Nova Biomedical.

Cunsolo, V., Distefano, S., 2010. From volunteer to cloud computing: cloud@ home. Conf. Comput. Front. 103–104.

Cunsolo, V., Distefano, S., Puliafito, A., Scarp, M., 2009. Cloud@ home: Bridging the gap between volunteer and cloud computing. ICIC'09 Proc. 5th Int. Conf. Emerg. Intell. Comput. Technol. Appl. 2009.

Cunsolo, V.D., Distefano, S., Puliafito, A., Scarpa, M., 2009. Volunteer computing and desktop cloud: The cloud@ home paradigm. In: Network Computing and Applications, 2009. NCA 2009. Eighth IEEE International Symposium on. IEEE, pp. 134–139.

Field, A., 2009. Discovering statistics using SPSS, Third. ed. SAGE Publications Ltd.

Garg, S.K., Versteeg, S., Buyya, R., 2013. A framework for ranking of cloud computing services. Futur. Gener. Comput. Syst. 29, 1012–1023.

Goiri, Í., Julià, F., Fitó, J.O., Macías, M., Guitart, J., 2012. Supporting CPU-based guarantees in cloud SLAs via resource-level QoS metrics. Futur. Gener. Comput. Syst. 28, 1295–1302.

Harutyunyan, A., Blomer, J., Buncic, P., Charalampidis, I., Grey, F., Karneyeu, A., Larsen, D., Lombraña González, D., Lisec, J., Segal, B., Skands, P., 2012. CernVM Co-Pilot: an Extensible Framework for Building Scalable Computing Infrastructures on the Cloud. J. Phys. Conf. Ser. 396, 032054.

Kirby, G., Dearle, A., Macdonald, A., Fernandes, A., 2010. An Approach to Ad hoc Cloud Computing. Arxiv Prepr. arXiv1002.4738.

Kondo, D., Taufer, M., Brooks, C., 2004. Characterizing and evaluating desktop grids: An empirical study. Int. Parallel Distrib. Process. Symp. 2004 00.

Lange, K., 2009. Identifying shades of green: The SPECpower benchmarks. Computer (Long. Beach. Calif). 95–97.

Lenk, A., Menzel, M., Lipsky, J., Tai, S., Offermann, P., 2011. What Are You Paying For? Performance Benchmarking for Infrastructure-as-a-Service Offerings. 2011 IEEE 4th Int. Conf. Cloud Comput. 484–491.

Li, Z., O'Brien, L., Zhang, H., Cai, R., 2012. On a Catalogue of Metrics for Evaluating Commercial

Cloud Services. … Int. Conf. 164–173.
Peterson, L., Muir, S., Roscoe, T., Klingaman, A., 2006. PlanetLab Architecture : An Overview.
Rasmussen, R., Trick, M., 2008. Round robin scheduling–a survey. Eur. J. Oper. Res. 617–636.
Schwiegelshohn, U., Yahyapour, R., 1998. Analysis of first-come-first-serve parallel job scheduling. Proc. ninth Annu. ACM … 629–638.
Stantchev, V., 2009. Performance Evaluation of Cloud Computing Offerings. 2009 Third Int. Conf. Adv. Eng. Comput. Appl. Sci. 187–192.
Van, H.N., Tran, F.D., Menaud, J.-M., 2010. Performance and Power Management for Cloud Infrastructures. In: 2010 IEEE 3rd International Conference on Cloud Computing. Ieee, pp. 329–336.
Villegas, D., Antoniou, A., Sadjadi, S.M., Iosup, A., 2012. An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds. 2012 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. (ccgrid 2012) 2, 612–619.
Weissman, J.B., Sundarrajan, P., Gupta, A., Ryden, M., Nair, R., Chandra, A., 2011. Early experience with the distributed nebula cloud. In: Proceedings of the Fourth International Workshop on Data-Intensive Distributed Computing. ACM, pp. 17–26.

Factors Influencing the Implementation of a Private Government Cloud in Saudi Arabia

Amal Alkhlewi, Robert Walters and Gary B. Wills
Electronics and Computer Science, University of Southampton, Southampton, U.K.
{aa3d12, rjw1, gbw}@ecs.soton.ac.uk

Keywords: Cloud Computing, Government Cloud, e-Government, Private Cloud.

Abstract: The government of Saudi Arabia is in the process of moving to e-government. This transition is hindered by the weakness of ICT infrastructure within Saudi government agencies. The development of a private government cloud is a solution for rapidly improving this infrastructure. An exploratory study is conducted to identify the factors that affect the implementation of such a private government cloud. An expert review has confirmed the ten factors suggested from an initial literature review and identified five additional factors.

1 INTRODUCTION

The use of Information and Communication Technologies (ICT) by governments to provide more efficient and effective services is increasing worldwide (Ndou, 2004). The purpose of e-government is to provide efficient government management of and access to information for citizens, thus enhancing service delivery (UN, 2014).

The different government agencies in Saudi Arabia are at varying levels of ICT maturity, which hinders the horizontal and vertical provision of e-government services (Alghamdi et al., 2014). They also reported that Saudi Arabia is lacking ICT in rural areas and there is insufficient integration among government organisations and their branches.

Cloud computing can be used to help governments quickly develop and strengthen their ICT infrastructure (Wyld, 2009); (Khan et al., 2011); (Tripathi and Parihar, 2011); (Zwattendorfer et al., 2013). It allows governments to uniformly supply e-government services, irrespective of the different maturity levels of different government agencies (Tripathi and Parihar, 2011).

2 LITERATURE REVIEW

2.1 e-Government in Saudi Arabia

Al-Nuaim (2011) notes that, while the Saudi government has the necessary assets to fund e-government, implementation is impeded by the slow growth of government services. Several other challenges and obstacles have been noted which hamper the full implementation of e-government in Saudi Arabia, including infrastructure, cultural and organisational factors. In her study of how effectively e-government had been implemented in Saudi Arabia, Al-Nuaim (2011) found that 8 of 21 (41%) ministries had not yet implemented the main features of an e-government web site. In addition, 10 ministries (45.4%) were completely or partially in the first stage (web presence); 3 ministries (13.6%) were in the second stage (one-way interaction); and 6 ministries had no online service at all. Alfarraj et al., (2013) noted that the Yesser e-government programme had changed its vision from offering electronic services to supporting infrastructure projects, particularly of government organisations, citing weakness in the public sector's infrastructure as a justification for the change in vision.

Alshehri et al., (2012) noted several "systemic barriers to e-government in Saudi Arabia, including IT infrastructural weakness in the government sector, lack of public knowledge about e-government, lack of systems that provide security

and privacy of information, and lack of qualified IT and government service expert personnel."

2.2 Government Cloud

Government clouds are seen as a new model for e-government (Liang, 2012); (Hodgkinson, 2012). Wyld (2009) suggests that the value of cloud computing has great appeal to governments due to the dynamic nature of IT demands and the difficult economic conditions many governments face.

Despite the benefits, there are many challenges and obstacles to using cloud computing in general, and to its use in e-government in particular. Researchers have found that the implementation of such projects in developing countries is more difficult than in developed ones (Schuppan, 2009), and that there are social groups who cannot partake of the benefits of e-government (Helbig et al., 2009).

3 DISCUSSION

Song et al., (2013) state that changes must be implemented in order to introduce cloud computing into an organisation. Yet there has not been any research to date into what changes need to be made for the introduction of cloud computing in Saudi Arabian government agencies to be successful. To help determine the factors that affect the implementation of a private government cloud, government IT experts' opinions were elicited on two questions:

RQ1: What are the factors that pose challenges to the implementation of a private government cloud in Saudi Arabia?

RQ2: What are the factors affecting the successful implementation of a private government cloud in Saudi Arabia?

3.1 Success Factors for Implementation of a Private Government Cloud

A literature review was conducted to answer the questions listed above. The review revealed that several obstacles need to be overcome when developing a private cloud for intergovernmental interaction in Saudi Arabia. By identifying these obstacles, it has been possible to propose ten success factors for the implementation of a private government cloud in Saudi Arabia, as shown in Figure 1.

Figure 1: Success Factors for the Implementation of a Private Government Cloud in Saudi Arabia.

3.2 Confirming the Factors

An exploratory study was conducted to confirm the proposed factors with the desk-based study, since there is no basis framework for a private government cloud to work with. To facilitate this study, experts were consulted to review and confirm the proposed factors. The objectives for this expert review were:

- To review the factors identified in the desk-based study to enhance the framework (i.e. add, delete and modify its components)
- To identify additional factors unique to the culture of Saudi government agencies that have not been identified previously by the literature.

4 THE EXPLORATORY STUDY AND ITS RESULTS

The factors proposed were evaluated by interviewing experts working on IT projects within Saudi government agencies. Experts were chosen for interview at this stage since the findings from such a sample have more credibility than those from a sample that includes non-experts (Bhattacherjee, 2012).

4.1 Expert Review Design

The review was based on semi-structured interviews with IT experts from Saudi government agencies.

This research method was chosen because it enables in-depth discussions and exploration to be conducted.

A person was considered an expert if they had at least five years' experience of working on IT projects within a Saudi government agency. These 12 IT experts were recruited from different government and semi-government organisations, and in different locations around Saudi Arabia. The interviews were conducted face-to-face, or over the phone, or online, based on the availability and location of the expert.

The interviews included both closed and open questions. The closed questions were concerned with obtaining the experts' opinions on the factors in the proposed framework. Experts were also encouraged to comment on the proposed factors. The open questions tried to identify further factors that had not been recognised in the desk-based study.

4.2 Expert Review Results

The first question asked of the experts was to give their opinion on the importance of the proposed factors. The second question, was to identify factors not mentioned in the study. The remaining questions were used to identify challenges and barriers to the implementation of a private government cloud in Saudi Arabia. The experts' opinions were analysed to produce the following results.

4.2.1 Review of Proposed Factors

There was consensus among the respondents that all the proposed factors were important except for two anomalies. Expert B did not find Top Management Support an important factor, stating that '*Usually this is not a factor to stop the project*'. Expert F did not consider Reliability and Business Process Re-Engineering to be important since '*Privately run clouds are more efficient than a government operated setup*' and '*where IT services are hosted is not relevant to the actual business processes*.'

4.2.2 Additional Factors

One question asked experts '*What other factors do you recommend to ensure the successful implementation of a private G-cloud?*' This question was intended to identify factors not mentioned in the proposed framework. The answers are summarised in Table 1.

4.2.3 Obstacles

Experts were asked to identify challenges to the implementation of cloud computing. The challenges identified were used to discover additional factors not mentioned in the proposed framework. The answers are also summarised in Table 1.

Table 1: Suggested Factors and Challenges.

Suggested Factors	Suggested Challenges
• Training for the IT-team • Data Knowledge and Quality management • Business Continuity Plan • Disaster Recovery Plan • Communication • Standards and frameworks to govern the cloud services provided • Documentation • Standards for information exchange • Project management office • Transparency	• Product limitations • Data Centre facilities preparation • Lack of local skills in Cloud Computing • Lack of local training facilities • Unrealistic schedules from management to complete projects • Security and Privacy • Interoperability and Portability • Reliability and Availability • Legal aspects • Compatibility with existing systems • Training staff

4.2.4 Expert Review Findings

It was clear that the proposed factors were considered to be unanimously important by the experts, all but Top Management Support, Reliability and Business Process Re-Engineering. One expert each did not consider of the previous factors to be important. Since the majority of the results were found to be in agreement with all the proposed factors, it was not found necessary to remove these factors.

Five additional factors were discovered by synthesising the expert' suggestions. These factors are: Communication, Standards for information exchange, Training for IT staff and end-users, Knowledge management, and Business continuity and disaster recovery plans. Other factors were suggested but were rejected, as they were included as part of the previously proposed factors. The updated factors are shown in Figure 2.

Figure 2: Updated Success Factors for the Implementation of a Private Government Cloud in Saudi Arabia.

5 CONCLUSION

This paper suggests that the implementation of a private government cloud will help strengthen the ICT infrastructure in Saudi government agencies. This will facilitate the Saudi government's e-government initiatives. A qualitative review of the literature identified ten success factors for the implementation of a private government cloud in Saudi Arabia. To confirm these factors an expert review of twelve IT experts from Saudi government agencies was conducted. The expert review confirmed the importance of the proposed ten factors and identified five additional ones. The next step will be to use triangulation to validate these factors.

ACKNOWLEDGEMENT

The authors acknowledge Jubail University College, an affiliate of the Royal Commission for Jubail & Yanbu, for funding this research.

REFERENCES

Alfarraj, O., Alhussain, T. & Abugabah, A., 2013. Identifying the Factors Influencing the Development of eGovernment in Saudi Arabia: The Employment of Grounded Theory Techniques.. *International Journal of Information and Education Technology,* 3(3), p. 319.

Alghamdi, I. A., Goodwin, R. & Rampersad, G., 2014. Organizational E-Government Readiness: An Investigation in Saudi Arabia. *International Journal of Business and Management,* 9(5), p. 14.

Al-Nuaim, H. A., 2011. An Evaluation Framework for Saudi E-Government. *Journal of e-Government Studies and Best Practices,* Volume 2011, pp. 1-12.

Alshehri, M., Drew, S. & Alfarraj, O., 2012. A Comprehensive Analysis of E-government services adoption in Saudi Arabia: Obstacles and Challenges. *International Journal of Advanced Computer Science and Applications,* 3(2), pp. 1-6.

Bhattacherjee, A., 2012. *Social Science Research: Principles, Methods, and Practices.* s.l.:Global Text Project.

Helbig, N., Ramón Gil-García, J. & Ferro, E., 2009. Understanding the complexity of electronic government: Implications from the digital divide literature. *Government Information Quarterly,* 26(1), pp. 89-97.

Hodgkinson, S., 2012. *Why government agencies need the cloud,* s.l.: OVUM.

Khan, F., Zhang, B., Khan, S. & Chen, S., 2011. *Technological leap frogging e-government through cloud computing.* Shenzhen , IEEE, p. 201–206.

Liang, J., 2012. *Government cloud: enhancing efficiency of e-government and providing better public services.* Shanghai, IEEE.

Ndou, V., 2004. E-government for developing countries: opportunities and challenges. *The Electronic Journal of Information Systems in Developing Countries,* 18(1), pp. 1-24.

Schuppan, T., 2009. E-Government in developing countries: Experiences from sub-Saharan Africa. *Government Information Quarterly,* 26(1), pp. 118-127.

Song, S.-h., Shin, S. Y. & Kim, J.-y., 2013. *A study on method deploying efficient cloud service framework in the public sector.* PyeongChang , IEEE.

Tripathi, A. & Parihar, B., 2011. *E-Governance challenges and cloud benefits.* Shanghai, IEEE.

UN, 2014. *UNITED NATIONS E-GOVERNMENT SURVEY 2014,* New York: United Nations.

Wyld, D. C., 2009. *Moving to the cloud: An introduction to cloud computing in government.* s.l.:IBM Center for the Business of Government.

Zwattendorfer, B., Stranacher, K., Tauber, A. & Reichstädter, P., 2013. *Cloud Computing in E-Government across Europe.* Berlin Heidelberg, Springer.

The Improved Cloud Computing Adoption Framework to Deliver Secure Services

Muthu Ramachandran[1], Victor Chang[1] and Chung-Sheng Li[2]
[1]School of Computing, Creative Technologies and Engineering,
Leeds Beckett University, Leeds LS6 3QS, U.K.
[2]IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A.
{m.ramachandran, v.i.chang}@leedsbeckett.ac.uk, csli@us.ibm.com

Keywords: Cloud Computing Adoption Framework Update (CCAF 1.1), Cloud Security, Framework for Cloud.

Abstract: This paper describes a high-level approach for our improved Cloud Computing Adoption Framework update 1 (CCAF 1.1), which emphasizes on the security policies, recommendations, techniques and technologies to be updated in our framework. Motivation, background, security overview and recent attack methods have been discussed. We propose a solution based on arising needs to improve current Cloud security, Fine Grained Security Model (FGSM) which is designed to integrate three different types of security methods and offer multi-layered security for a better data protection. Technologies and techniques behind FGSM have been explained and will be useful for our CCAF 1.1 development.

1 INTRODUCTION

Cloud Computing has transformed many organizations in several ways. First, organizations can consolidate the infrastructure, since the deployment of virtual machines can replace the use of physical machines. While there are less computers, people and spaces being used, this helps organizations reduce the operational costs in the long-term. An alternative for small and medium businesses is to outsource their services to other vendors to reduce costs (Khajeh-Hosseini et al, 2010; Weinhardt et al., 2009;). Second, less carbon and wastes will be produced due to the scale down of servers, air-conditioning systems and spaces. In this way, Cloud Computing supports Green IT and sustainability to cut down energy and resource wastes (Khajeh-Hosseini et al., 2010; Marston et al., 2011). Third, Cloud Computing can streamline business processes at some organizations. For example, it takes less time and effort to find goods, package and deliver for supply chain service providers when orders have been received. This improves their work efficiency, since some operational tasks can be completed quicker with better (Marston et al., 2011). Fourth, Cloud Computing offers offers companies more business opportunities since they can work as service providers and can access wider groups of customers based in different parts of the country or the world (Weinhardt et al., 2009; Marston et al., 2011). Fifth, Cloud Computing can provide a platform for scientists and developers to use and share their code (Velte et al., 2009). They make use of libraries and APIs to directly interact on the Cloud. However, there are challenges such as security, data ownership and bottle neck to performance and services (Armbrust et al., 2010). Apart from all these challenges, different organizations have used Cloud Computing for different purposes. For example, Company A uses Cloud Computing for outsourcing since they outsource their servers to the vendors. Company B uses Cloud Computing to facilitate their demanding services. So at their peak hours, they use Cloud Computing to share the workload so that more tasks or requests can be completed quickly. Company C uses Cloud Computing to improve work efficiency by completing more workloads at the same time and they can reduce resources including human resources. Company D uses Cloud Computing to store all their experimental data in the Cloud so that they can use it whenever they have access to the internet. Company E use Cloud Computing so that all their office documents and orders are completed, processed and stored in the Cloud and they work as a mobile office as a service. Company F offers Cloud Computing as a Consulting

as a Service to help their clients develop infrastructure, platform and software according to their clients' need. Although security challenge applies in these six companies, the challenges that all six companies are facing, will require processes, recommendations and guideline to help them achieve their goals and objectives. In other words, they need a well-structured, proven and well-established framework to guide and help them achieve their goals, improves their efficiency, increase their business opportunities and teamwork, reduces errors and rate of failures. The development of a framework that takes challenges and resolution into considerations is highly recommended and should always be encouraged.

1.1 Overall Discussion about Cloud Computing Adoption Framework

There are researchers attempting to illustrate the framework approach for Cloud Computing best practices. Low et al (2011) describe how their Technology, Organization and Environment framework can be used and developed as their Cloud adoption framework. They used qualitative approach and sent out questionnaires to directors and decision-makers in Taiwanese firms. Based on their analysis, they validate their hypotheses. However, such an approach appears to be applicable to Taiwan and their proposal is not entirely adopted by other organizations in other countries. Khajeh-Hosseini et al (2010) present a case study and demonstrate a work similar to a framework level. They explain the strengths and weakness of adopting Cloud Computing and ways to reduce costs and improve efficiency. However, their work is not a framework addressing specific and general problems. They do not have comprehensive guidelines to help organizations at different levels of adoption rather than focusing on calculations of cost-involved in Cloud Computing adoption.

IBM (2010) has developed their IBM Cloud Adoption Framework to advise the best approaches and recommendations while developing services in different types of Clouds at the time of publication. They use diagrams to illustrate their concepts. However, there is a lack of real-life case studies to support their vision and points of views. This explains why a collaboration with independent researchers is helpful for Cloud Computing research. Chang and Li (2012) et al have started the first collaboration to demonstrate the first prototype of Financial Software as a Service (FSaaS) and illustrate FSaaS can be ported to different types of Clouds with its performance benchmark tested. More research outputs have been updated from Year 2012 onwards. Chang et al (2013 a) and Chang (2015) propose their Cloud Computing Business Model (CCBF) which has four major components and compiles a summary of successful deliveries and case studies of Cloud Computing. There are reported added value and benefits from organizations that have adopted Cloud Computing under the guidelines of CCBF. Selected results have been presented in their papers. However, there is no detailed information from the design to implementation to service delivery. Due to this reason, the next phase of work known as Cloud Computing Adoption Framework (CCAF) has been developed (Chang et al, 2013 b; Ramachandran and Chang, 2014). CCAF emphasizes more on the practical implementation, service delivery and resolution of problems rather than presenting the conceptual framework. There are detailed case studies in healthcare (Chang, 2014 a) and finance (Chang, 2014 b) to explain the process of transforming theory into practice, since service delivery with real users in place was a priority. However, there is a lack of demonstrations on security (despite of their three workshop papers), which is an important aspect of Cloud Computing service to ensure all services are well-protected.

In other words, the current version of CCAF needs revision by updating the security guidelines and business context. The emphasis should be as follows. First, how to make theory into practice. Several security papers have emphasized very much on the theoretical development and there is a lack of details describing how to reproduce similar results and replicate the success of delivering security services. Second, security technologies, measures and policies should be easily integrated with the existing practices. Third, the business context will be emphasized, since the improved framework should be adopted by industry and businesses that aim for long-term benefits such as cost reduction, business opportunities, profitability, improvement in efficiency and customer satisfaction as discussed in Section 1. The development of security and business solutions should be clear and easy to adopt. Thus, these three main factors drive us into the development of Cloud Computing Adoption Framework Update 1 (CCAF 1.1). Proof-of-concepts will be demonstrated to support our proposed CCAF 1.1.

1.2 The Integrated Data Center for Everything as a Service

To blend and manage security and business solution into CCAF 1.1, strategic directions have to be set and deployed to ensure that all future and emerging services, or Everything as a Service (EaaS), can be successfully delivered. EaaS includes design, deployment and guideline for Infrastructure, Platform and Software as a Service. Other value added services such as Business Process, Security and Consulting as a Service are also part of EaaS.

The rationale for the IBM's approach is to start with the next-generation data center. The aim is to consolidate all resources and improve the percentage of resource utilization. This can ensure that Data center can be fully used and not to waste much energy and space. Similarly, platform and software as a service can be built on top of a smart data center into an integrated system model (Li, 2014). The integration starts from the infrastructure as a service level where the server, storage, networks and system management software is pre-integrated prior to shipping to the data center. The scope of the pre-integration varies from single rack systems within a traditional data center to a full size datacenter-in-a-box container. All the hardware integration is important for EaaS, since it will take much less time to send the network from one end to the other within the data center. Performance and response time can be enhanced significantly. The downtime caused by the bottleneck of network and storage will be less likely to happen, since the integrated data center can provide intelligent systems to warn the system manager, reassign extra demands to under-utilized data centers and ensure all resources can be smartly utilized.

2 SECURITY UPDATES

This section describes security update for Cloud Computing Adoption Framework Update 1 (CCAF 1.1). Topics include cyber attacks overview and recent attack methods, which help revise the counter-attack and remedy actions or CCAF 1.1.

2.1 Security Overview

The data leakage incidents due to various reasons, as reported by the DataLossDB.org have been on the rise in recent years according to DataLossDB.org survey (2013). The rapid jump from 2005 to 2006 is due to various disclosure legislations. The term Threat can be divided into *Internal Threats*, and *External Threats*. The former is originated from authorized users compromising and exploiting internal systems, while the latter are from external attackers. In both cases, the attackers seek to compromise systems by accessing data, gaining control of systems and applications, or disrupting their operation. Based on the technical report (Li, 2014), 57% of the loss incidents are due to external attacks while 36% are due to insiders as of the end of July, 2013 based on the IBM survey.

To expand this area further, the *Internal Threats* can be further subdivided into threats from *Insiders with Malicious Intent*, and threats from *Unintentional Insiders*. The risk posed by a malicious insider intents on compromising internal systems must be mitigated by a range of security measures, including background checks, restricting access, physical monitoring, platform integrity monitoring and controls on desktop applications and operations as well as profiling and auditing of user interactions with key applications and data. With the threat landscape so defined, the primary threats that require mitigation include:

1. *Malcode*: This threat comes from programs, scripts, or macros that are malicious in nature and can execute on user machines. This category of threats is often subdivided into *viruses* and *Trojans*. A *virus* is code that is attached to or contained within a legitimate application or document. A *Trojan* is a program that has an externally visible purpose and behavior, but also has covert, malicious behavior that is invisible to the user. A variety of stealth technologies can be deployed to keep malcode installed without detection (e.g. root kits). Self-propagating code is also often referred to as a *Worm*.
2. *Vulnerabilities*: These are deficiencies in legitimate code running on internal computer systems. If an attacker can interact with a vulnerable system that is internal to a network, or provide data to it, then it is possible for the attacker to exploit such a vulnerability to compromise the system. As with malcode, the vulnerability threat has several sub-categories, for example, SQL injection and Cross Site Scripting vulnerabilities (XSS). The most devastating types of vulnerabilities are those designated as *Remote Code Execution*. These vulnerabilities can allow code execution natively on the computer containing the vulnerable code (for example, using browsers or browser plug-ins). During the week of April 6, 2009 alone, US-CERT reported 142

vulnerabilities rated high or medium value.

3. *Data Loss and Leakage*: This threat often comes from insiders unintentionally transferring restricted information to external systems. This can also result from malcode installed on users' machines. Detecting and preventing the transfer of sensitive information from within an organization to an unauthorized external site is the focus. Data loss can also result from the intentional actions of insiders focused on stealing valuable information.
4. *Denial of Service* (DOS): This threat comes from external users or systems attacking a targeted system's infrastructure with the intent to disrupt its operation to the degree necessary to degrade or disable its ability to serve its users. There are various forms of DOS attacks: one is the vulnerability DOS; some are vulnerabilities that might not be exploitable to gain Remote Code Execution, but can be exploited to crash the system. More common are DOS disruptions that arise from a high volume of spurious (attacker) traffic that overwhelms a network or host computer. If an attacker can construct a sequence of packets that overloads a host computer's capacity, then a flood of these packets can cause a denial of service. *Bandwidth DOS* attacks also seek to exhaust the network capacity by flooding the network with traffic. Often these attacks are coordinated to originate from thousands of different host computers (Distributed Denial of Service Attack) that have been compromised with botnet malcode installed covertly. These threats are unleashed by attackers with increasing creativity, for example: malcode often communicates over encrypted sessions; Javascript is often used to evade Intrusion Prevention Systems by obfuscating exploits; low bandwidth data leakage is difficult to detect and stop on the wire.
5. *Web Vandalism and Propaganda*: Attacks that deface Web pages, or spread political messages to anyone with access to the Internet.
6. *Botnets*: Collections of compromised computers (i.e. zombie computers) running programs, such as worms, Trojan horses, or backdoors, under a common command and control structure.
7. *Equipment Disruption*: This is the threat of physical tampering or destruction of computing equipment. For example, military activities that use computers and satellites for coordination are at risk from this type of physical attack.
8. *Critical Infrastructure Attack*: National electric power, water, fuel, communications, commercial and transportation systems are all vulnerable to cyber attacks.

2.2 Recent Attack Methods

Understanding the recent attack methods will help revise the guidelines and software fixes for CCAF 1.1. There is a list of cyber security incidents between February and Augst of 2011 compiled by X-Force of IBM, which include Amazon's loss of data in 2011 and 2012, and the problems with Elastic Load Balancing services in 2013 and RSA's hacked data and services (Li, 2014). It is apparent that the frequency and the size of the impact monotonically increased during this period.

Among all these incidents, the most severe incident is the attack on RSA during March 2011. This incident involves what is known as ***Five-layered of Advanced Persistent Threat*** (APT), and often includes the following five phases over an extended period of time:

1. **Social Engineering:** Initially, spear phishing emails were sent over a two-day period to small groups of employees with RSA. The email subject line read *2011 Recruitment Plan*, was from beyond.com – an HR partner firm of RSA. The spreadsheet contained a zero-day exploit that installs a backdoor through an Adobe Flash vulnerability. One of the RSA employees clicked the attachment from junk mail.
2. **Back Door:** The malware installed a customized remote administration tool known as Poison Ivy RAT to allow external control of the PC or server, and set up the tool in a reverse-connect mode
3. **Moving Laterally:** The malware first harvested access credentials from the compromised users (user, domain admin, and service accounts), then performed privilege escalation on non-administrative users in the targeted systems, and then moved on to gain access to key high value targets.
4. **Data Gathering:** Attacker behind the malware in the RSA case established access to staging servers at key aggregation points.
5. **Exfiltrate:** The attacker then used FTP to transfer many password-protected RAR files from the RSA file server to an outside staging server on an external, compromised machine at a hosting provider. Once the transfer completed, the footprints were wiped clean

making it impossible to trace back to the attacker(s).

3 OUR PROPOSED SOLUTION

This section describes our proposal for designing and deploying the security solutions. The approach is to use a framework that can integrate different aspects of security. We propose the "Fine Grained Security Model" (FGSM), which offers the multi-layered security layer for Cloud Computing services. Since each type of security has its strengths and weaknesses, the combination of different security solutions can enhance the strengths and reduce the weakness if only one single solution is deployed.

3.1 The Overview

Before introducing the details of our updated framework, each element of the CCAF security is described as follows.

Identification is a basic and the first process of establishing and distinguishing amongst person/user & admin ids, a program/process/another computer ids, and data connections and communications.

Privacy is the key to maintaining the success of cloud computing and its impact on sharing information for social networking and teamwork on a specific project. This can be maintained by allowing users to choose when and what they wish to share in addition to allowing encryption and decryption facilities when they need to protect specific information/data/media content.

Integrity is defined as a process of maintaining consistency of actions, communications, values, methods, measures, principles, expectations, and outcomes. Ethical values are important for cloud service providers to protect integrity of cloud user's data with honesty, truthfulness and accuracy at all time.

Durability is also known as, persistency of user actions and services in use should include sessions and multiple sessions.

The other important aspects are as follows.

Confidentiality, Privacy and Trust – These are well known basic attributes of digital security such as authentication and authorization of information as well protecting privacy and trust.

Cloud Services Security – This includes security on all its services such as SaaS, PaaS, and IaaS. This is the key area of attention needed for achieving cloud security.

Big Data Security – This category is again paramount to sustaining cloud technology. This includes protecting and recovering planning for cloud data and service centers. It is also important to secure data in transactions.

Physical Protection of Cloud Assets – This category belongs to protecting cloud centers and its assets.

3.2 The Fined Grained Security Model

CCAF security software implementation is demonstrated by the use of the Fine-Grained Security Model (FGSM), which has layers of security mechanism to allow multi-layered protection. This can ensure reduction in the infections by trojans, virus, worms, and unsolicited hacking and denial of service attacks. Each layer has its own protection and is in charge of one or multiple duties in the protection, preventive measurement and quarantine action presented in Figure 1.

All the features in FGSM include access control, intrusion detection system (IDS) and intrusion prevention system (IPS), this fine-grained security framework introduced fine-grained perimeter defense. The layer description is as follows.

- The first layer of defense is **Access Control and firewall** to allow restricted members to access.
- The second layer consists of the **IDS and IPS**. The aim is to detect attack, intrusion and penetration, and also provide up-to-date technologies to prevent attacks such as DoS, anti-spoofing, port scanning, known vulnerabilities, pattern-based attacks, parameter tampering, cross site scripting, SQL injection and cookie poisoning. The identity management is enforced to ensure that right level of access is only granted to the right person.
- The third layer, being an innovative approach, **Encryption**, enforces top down

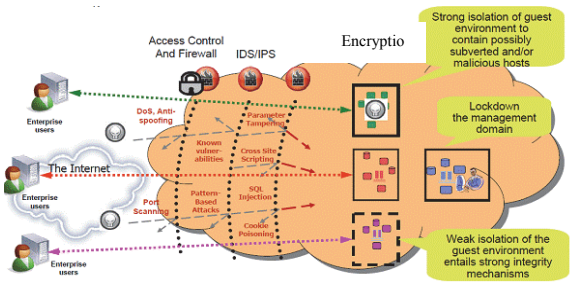

Figure 1: The Fine-Grained Security Model offered by CCAF.

policy based security management; integrity management. This feature monitors and provides early warning as soon as the behavior of the fine-grained entity starts to behave abnormally; and end-to-end continuous assurance which includes the investigation and remediation after an abnormality is detected.

3.3 Technologies behind FGSM

This section describes the technologies behind FGSM, which uses XACML 3.0 (Extensible Access Control Markup Language), an XML-schema to define the which ports for secure communications with respect to the IP addresses. All the ports support secure ssh and ftp. XACML 3.0 has followed the industry standard to define the access control policy and how to access requests based on rules supported by the policies (Parducci et al., 2013). Our scripts have been carefully reviewed and tested under the testing and live environments. Additionally, the use of the integrated hardware and software technologies ensure a better protection for users and organizations. The description for each security layer is as follows.

In the first layer, firewall, we adopt the combination of Cisco and XACML technologies. Cisco routers and networking infrastructure allow us to set the firewall and monitor any abnormal activities. The use of XACML can enforce the strength of the security and minimize any errors, which include acknowledging the malicious (but well-hidden) code as the safe code.

In the second layer, identity management defines the type of users and their privilege and permission. These include the followings:

- Users: who can encrypt each key from his block and his own key. This step is to ensure that all the data that users access and store are protected in the Cloud.
- CCAF server: Three functions are as follows. First, it can authenticate users during the storage and retrieval process. Second, it offers access control for users. Third, it encrypts data between users and the Cloud.
- Security Manager (SM): This stores metadata which includes block signatures, encrypted keys and process identity management check. SM also checks whether a user is authorized to retrieve a file that he/she has requested, which offers an additional access control.

In the third layer, it adopts convergent encryption, which aims to consolidate all the files to be encrypted for storage. There are advanced but easy-to-use cryptography algorithms deployed. We can minimize the de-duplication of the same files and can monitor the changes and updates of encrypted files. This can ensure all the data coming in and out of the CCAF server to be protected to reduce the possibility that messages to be hijacked.

3.4 Isolation and Quarantine

The FGSM also provides the detection and intrusion systems which record the typical behaviors of the trojans, viruses and worms. When the identified trojans, viruses and malicious code are found, they are isolated and sent to the quarantine area immediately. The strong isolation and integrity management are jointly used to protect user safety. Strong isolation is used to detect vulnerabilities in any of the cloud services, including the block of unauthorized IPs and attack points/ports. Quarantine is the next step to enforce security. It first backups the data safely and then attempts to quarantine infected data. If a quarantine action is unsuccessful, it informs the system architect. The files can be kept under “quarantine area” or chosen to be deleted.

3.5 Resilient Computing

As discussed in Section 1, the intelligent Data Center will integrate all hardware infrastructure and applications supporting the hardware. The benefit is to provide a better access, hardware-software integration and performance than the current Data Center deployment. With regard to this, IBM has proposed the Resilient Computing which integrates Cloud Computing hardware and software with security. The updated CCAF framework will be essential to IBM Resilient Computing development.

3.6 Discussion

This paper describes the rationale and methodology of our CCAF framework, in which the FGSM is at the center of the illustration tro validate and demonstrate our approach and solution for security. Large scale experiments and testing results have been undertaken and discussed in our transactions papers, in which performance results and pentrating testing had been used to test how robust the FGSM system could offer (Chang and Ramachandran, 2015). This paper is focused on the system design for Emerging Software as a Service and Analytics and not on the empirical results with their

discussions. It also reviews the previous work for the development framework and proposes the requirements for the next phase, CCAF Version 2.

4 CONCLUSION AND FUTRUE WORK

This paper provides a strategic overview and direction for the improved Cloud Computing Adoption Framework update 1 (CCAF 1.1), in which the emphasis is on the update on security policy, technologies and techniques used. The security recommendation and updates can help organizations building and offering better protected services. Different types of technologies and techniques have been discussed. The proposed Fine Grained Security Model (FGSM) offers multi-layered security and is a suitable solution in the deployment of Cloud Computing services, since each single solution has its weakness. The core technology in each layer of FGSM have been described and justified, which includes the firewall, the identity management and convergent encryption. The combination of three main security solutions in FGSM can enforce security service.

The FGSM prototype will be developed and then thoroughly tested in the laboratory conditions. We plan to use ethical hacking and penetration testing approached to test the robustness of our FGSM security. This will be fully implemented in our CCAF and eventually the development of Resilient Computing. If the results are positively in favor of our prototype and security strategy, we will update our recommendation, results and guidelines, which will be developed into CCAF Version 2.

REFERENCES

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M., 2010. A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58.

Chang, V., 2014 a. Cloud Computing for brain segmentation – a perspective from the technology and evaluations. International Journal of Big Data Intelligence, 1, (4), 192-204.

Chang, V., 2014 b. The business intelligence as a service in the cloud. *Future Generation Computer Systems*, *37*, 512-534.

Chang, V., 2015. A Proposed Cloud Computing Business Framework, ISBNs: 9781634820172 (print), Nova Science Publisher.

Chang, V., Li, C. S., De Roure, D., Wills, G., Walters, R. J., & Chee, C., 2012. The financial clouds review. Cloud Computing Advancements in Design, Implementation, and Technologies, 125.

Chang, V., Walters, R. J. & Wills, G., 2013 a. The development that leads to the Cloud Computing Business Framework. International Journal of Information Management, June, 33, (3), 524-538.

Chang, V., Walters, R. J. & Wills, G., 2013 b. Cloud Storage and Bioinformatics in a private cloud deployment: Lessons for Data Intensive research. In, Cloud Computing and Service Science, Springer Lecture Notes Series, Springer Book.

Chang, V. & Ramachandran, M., Towards achieving Big Data Security with the Cloud Computing Adoption Framework, IEEE Transactions on Services Computing, forthcoming.

DataLossDB.org survey, 2013, accessible on http://datalossdb.org/us_states in 2013.

IBM, 2010. Defining a framework for cloud adoption, technical report.

Khajeh-Hosseini, A., Greenwood, D., & Sommerville, I., 2010, July. Cloud migration: A case study of migrating an enterprise it system to iaas. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on (pp. 450-457).

Li, C. S, 2014. Resilient Computing, technical report, IBM.

Low, C., Chen, Y., & Wu, M., 2011. Understanding the determinants of cloud computing adoption. *Industrial management & data systems*, *111*(7), 1006-1023.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A., 2011. Cloud computing—The business perspective. *Decision Support Systems*,*51*(1), 176-189.

Parducci, B., Lockhart, H., Levinson, R., 2013. OASIS eXtensible Access Control Markup Language (XACML) TC, technical report, accessible on https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

Ramachandran, M., & Chang, V. 2014. Financial Software as a Service–A Paradigm for Risk Modelling and Analytics. *International Journal of Organizational and Collective Intelligence* 4(3).

Velte, T., Velte, A., & Elsenpeter, R., 2009. *Cloud computing, a practical approach*. McGraw-Hill, Inc.

Weinhardt, C., Anandasivam, D. I. W. A., Blau, B., Borissov, D. I. N., Meinl, D. M. T., Michalk, D. I. W. W., & Stößer, J., 2009. Cloud computing–a classification, business models, and research directions. Business & Information Systems Engineering, 1(5), 391-399.

Towards an Integrated Conceptual Model for Cloud Adoption in Saudi Arabia

Nouf Alkhater[1], Victor Chang[2], Gary Wills[1] and Robert Walters[1]
[1]*Electronics and Computer Science, University of Southampton, Southampton, U.K.*
[2]*Leeds Beckett University, Leeds, U.K.*
{nrma1c12, gbw, rjw1}@ecs.soton.ac.uk, v.i.chang@leedsbeckett.ac.uk

Keywords: Cloud Computing, Adoption, Factors, TOE Framework.

Abstract: There are several advantages of utilising cloud computing in organisations such as cost saving and flexibility in acquiring resources. The use of cloud computing in developing countries, such as Saudi Arabia, is still in its early stages and has not been as widely adopted there as in developed countries. In fact, moving a current system to the cloud depends on many factors that may affect a Saudi Arabian organisation's decision to adopt the cloud. In order to encourage the adoption of cloud technology it is essential to understand why some enterprises are more prepared than others to move to the cloud. Hence, the aim of this research is to examine factors that might impact on a Saudi Arabian organisation's intention to adopt cloud computing. In this paper, we propose a conceptual model which integrates aspects of the Technology Organisation Environment (TOE) framework. The proposed model identifies the key factors that might influence organisations to employ cloud services. Our findings show that all the proposed factors in the cloud adoption model, except for competitive pressure and trading partner pressure, are statically significant.

1 INTRODUCTION

Cloud computing is the emerging paradigm of delivering IT services to end users as a utility service over the Internet. A number of technologies are used to make cloud computing happen, including virtualisation and Web 2.0, and their presence makes cloud computing more efficient and usable (Jeffery and Neidecker-Lutz, 2010). The concept of cloud computing started in the 1960s, but the expression "cloud computing" became widely popular only in 2007 (Chen et al., 2010). A number of different proposals, such as grid computing, have been developed but none of them has achieved cloud computing's level of success in offering services to the general public.

Cloud computing can bring several advantages to organisations. The cloud can reduce capital expenditure for both large and small organisations and enable them to pay only for the services they consume rather than setting up in-house IT infrastructure (Buyya et al., 2009). Cloud computing offers business opportunities and flexibility for organisations to increase their revenues (Marston et al., 2011). Despite all these benefits, some organisations hesitate to migrate their work to the cloud. To help organisations achieve their long-term goals, a number of frameworks have been developed to provide guidelines and recommendations for cloud adoption, such as Chang et al., (2013) and Chang (2015).

An interesting observation about the proposed models and frameworks in previous studies is that they focus on the costs and benefits of cloud adoption. Furthermore, there is a lack of empirical studies conducted to examine the influential factors for adopting cloud technology at enterprise level (Low et al., 2011; Borgman et al., 2013). Additionally, all these adoption cases have focused on deployment cases in the West; the adoption rate in the Saudi Arabia is in the beginning phase. Hence, the aim of this study is to carry out an in-depth investigation of factors that influence an enterprise's decision to use cloud technology in Saudi Arabia. An integrated conceptual model has been proposed in order to identify what could drive an organisation to use cloud services or prevent them from doing so.

The structure of the paper is as it follows. Section 2 begins with the background of cloud computing and then provides a critical review of the

existing work and theories in order to identify factors that affect an organisation's decision to adopt cloud computing. Section 3 presents the conceptual model for cloud adoption in Saudi Arabia. The research methodology is discussed in Section 4. Section 5 provides the preliminary results. Finally, the summary and future work are presented in Section 6.

2 LITERATURE REVIEW

2.1 Benefits of Cloud Migration

This section presents the benefits for organisations that adopt cloud computing. First, cloud computing offers cost reductions and savings due to the outsourcing of hardware and services. Organisations can save on operational costs in that they no longer have to buy machines, provide a bigger space for storage, and pay upgrade costs and staffing costs (Chang, 2015). The responsibilities and costs involved in improving and upgrading systems are managed by the cloud service providers (Armbrust et al., 2010; Buyya et al., 2009; Jeffery and Neidecker-Lutz, 2010). Secondly, cloud technology provides an opportunity for organisations to scale their services easily and tailor these to specific needs. For example, customised functions can be designed for the company staff so that they can perform their tasks quickly and easily. Thirdly, cloud computing supports green IT since the costs of buying and maintaining servers are reduced with fewer carbon emissions and less energy consumption taking place (Buyya et al., 2012; Marston et al., 2011). Additionally, enterprises can design, build and run their applications more smoothly, since they can be tested in virtual machines as many times as they like. Finally, the flexibility of delivering computing services can drive organisations to migrate their services to the cloud (Foster et al., 2008).

2.2 A Review of Proposed Approaches to Cloud Migration

This section reviews existing work and models related to the migration to cloud computing in order to explore how far the security issues are considered in them.

First, Khajeh-Hosseini et al., (2010) reported a case study that refers to a legacy migration of system in the gas and oil sector. This study examined the migration of an IT system from an enterprise data centre to Amazon's EC2. The cost analysis of the company is presented. In addition, the case study presents the possible advantages and risks linked to the migration of the system based on the point of view of managers and other staff, except the security manager and other security experts. In fact, the most important views that need to be taken into account for migration process are those of the security staff. Their findings indicate that the use of cloud infrastructure will decrease the enterprise's costs. Their results are also useful for decision-making purposes as they will help analysts to find solutions to upcoming issues associated with the adoption of a cloud by enterprises. However, their work does not take into account the security aspect. In fact, security is a vital factor in cloud migration and it needs to be considered as an essential element in the migration process.

Khajeh-Hosseini et al., (2011a; 2011b) extended their previous study to develop a toolkit that helps decision-makers and organisations address their concerns during the migration process; the toolkit provides a framework that can be used to evaluate the migration of businesses from a enterprise data centre to a public cloud. The first tool consists of a list of questions; this helps enterprises to determine whether a public cloud is a suitable technology for their IT system. The second tool is helpful for the decision-makers in terms of estimating the costs of employing a public cloud. Their third tool is a spreadsheet that demonstrates the possible risks and benefits associated with a public cloud from a general organisational perspective. Their evaluation of the tools based on different case studies focuses only on the cost model. Indeed, the proposed methods are a good starting point for risk assessment and are useful for decision-makers as they cover some issues regarding migration to a public cloud. However, this work only considers the cost of the infrastructure when using one type of cloud (the public cloud).

Klems et al., (2009) proposed a framework to measure the costs of using IT infrastructure in the cloud. They compared it with conventional IT approaches, such as the cost of setting up in-house IT infrastructure or a grid computing service. They dealt with costs in their framework under direct and indirect costs. IT infrastructure resources are an example of direct costs, whereas an indirect cost is incurred by the failure to meet business goals and set up training courses for the new technology. The framework was evaluated based on two case studies. However, their work was in the development phase and therefore the results are not provided. Also, this study did not consider the security aspect.

Hajjat et al., (2010) proposed a model for the migration of an enterprise's applications to a hybrid cloud. The aim of this study was to identify the costs and benefits of migrating part of the system to the cloud. The effectiveness of this approach was briefly evaluated based on a case study of the migration of applications to the cloud. However, this work does not mention how the cost can be computed and only focuses on one type of cloud (the hybrid cloud). They also did not consider the security aspect.

Hu and Klein (2009) have carried out a study to investigate privacy issues during migrating e-commerce applications to the cloud. Their study suggests that the user's data and critical business information must be encrypted during the migration process. The authors have also studied and compared existing data encryption methods in different layers (storage, database, middleware and application). They argue that the middleware layer encryption is the most effective approach for migrating e-commerce applications to the cloud in terms of performance. The evaluation of their work was based on a case study for an e-marketplace application. Indeed, this approach discussed data encryption, particularly for the transmission of e-commerce applications to the cloud. This method helps to ensure privacy of data and provides protection for applications during the migration process. Nevertheless, the authors did not point out how the data and applications would be migrated to the cloud; they also ignored the other aspects of security and privacy that need to be considered.

Hao et al., (2009) proposed a cost model that can be used to determine the type of services included in migration and their possible location. The model that they developed used a genetic algorithm to provide an effective decision for service migration, by looking for the most optimal migration decisions. In this study, besides considering the cost of service migration, they evaluated the cost of consistency maintenance and communication. It is important to have strong decision support for the infrastructure support, prior to migration. However, the authors omit security in the migration process and they deal only with the security aspect that involves accessing the control process by proposed mutual authentication using certificate authority.

Kaisler and Money (2011) have conducted a study to investigate issues associated with service migration to a cloud, as well as the security problems involved with service implementation. They considered several security challenges. It is noticeable that this study simply lists the possible challenges without any evaluation; it also ignores the security aspects in the migration process.

2.3 A Review of Proposed Models for the Adoption of New Technologies

This section describes relevant theories and frameworks for the adoption of new technologies. It includes the TOE framework, the Diffusion of Innovations (DOI) theory and the institutional theory, which have been widely adopted by researchers.

Tornatzky and Fleischer (1990) proposed the TOE framework to analyse the acceptance of new IT technologies at an organisational level. The TOE framework investigates the impact of three factors, Technology, Organisation and Environment, on the organisation's decision to adopt a new technology. According to Tornatzky and Fleischer (1990) and Chau and Tam (1997), TOE can be summarised as follows:

- The technology aspect describes the internal and external characteristics of the new technology and how adopting a new technology can influence the organisation.
- The organisational context is focused on different measures that can influence the direction of the organisation, for example, firm size and scope of interests.
- The environmental context refers to the characteristics of the environment where an organisation operates its business and might have a significant impact on their decision. Government regulation and competitors are an example of the environmental context.

The DOI was proposed by Rogers (1995). DOI is a widely used theory in information system research to examine user acceptance of new ideas and technologies. The DOI theory presents five attributes that have a direct influence on adoption rate: relative advantage, complexity, compatibility, trialability and observability.

The institutional theory is one of the common theories usually used for explaining the adoption of IT technologies (Scott and Christensen, 1995; Scott, 2001). The different between the TOE framework and institutional theory is that institutional theory contains two important elements (trading partner pressure and competitors) in the environmental context of the TOE framework which might play an important role in an organisation's decision to adopt new technologies.

The other models which have been built based on previous theories in order to identify the factors that affect on a firm's decision to implement cloud computing are presented in a previous work

(Alkhater et al., 2014).

3 CONCEPTUAL MODEL

As discussed in Section 1, some of the proposed frameworks and models do not fully address the in-depth investigations on what factors influence cloud adoption for organisations. The TOE framework has been widely adopted and is a suitable model for improvement since it has a proven track record of successful integration (Tornatzky and Fleischer, 1990; Chau and Tam, 1997). Additionally, another benefit of using the TOE framework is that this framework predicts and examines the adoption of technology based on three aspects: technology context, organisation context and environment context.

In this paper, an integrated model has been proposed to identify factors that impact on an enterprise's intent to adopt the cloud services in Saudi Arabia. The initial model has been constructed by integrating aspects of the TOE framework and combining the most important factors from the DOI theory and institutional theory along with other factors (trust, privacy and physical location) that have not yet been investigated in any previous studies as main factors that may have an impact on the organisation's decision to adopt cloud services. The conceptual model for cloud computing adoption in Saudi Arabia is presented in Figure 1. Moreover, Table 1 identifies factors involved in the cloud adoption model; the details of these proposed factors were discussed in a previous work (Alkhater et al., 2014).

Table 1: The factors identified for cloud adoption.

Factors	Sub-dimensions
Technological Factors	Availability
	Reliability
	Security
	Privacy
	Trust
	Relative advantage
	Compatibility
	Complexity
Organisational Factors	Top management support
	Organisation size
	Technology readiness
Environmental Factors	Compliance with regulations
	Competitive pressure
	Trading partner pressure
	Physical location

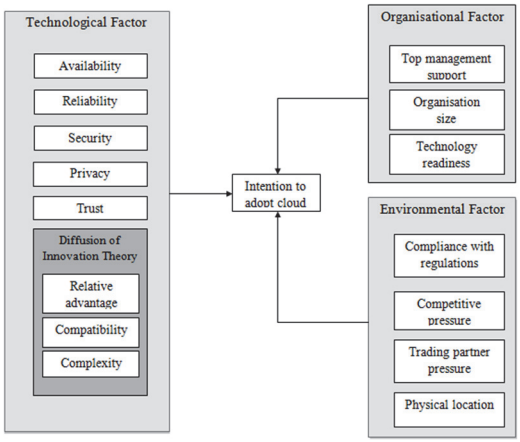

Figure 1: A conceptual model for cloud computing adoption.

4 METHODOLOGY

An expert review is a simple method that enables researchers to collect data from experts who have knowledge of the topic under study. This technique can be used in quantitative, qualitative or mixed methods at different stages of the study (Tessmer, 1993). In this initial study, semi-structured interviews were used for collecting data from twenty IT experts working in IT departments in different Saudi organisations. The study population includes IT staff or managers. The aim of the interviewing IT experts was to review factors that were previously identified in Section 3. A second objective was to discover other factors left unstated in former studies. The interviewees in this study were working in various sectors, such as petrochemicals, oil and gas and engineering, in large organisations and small and medium-sized enterprises with at least five years' working experience in IT. Seven of the participants in this study were working in companies that had already adopted cloud computing, while thirteen (65%) of them were not.

5 RESULTS

This section shows the results of this preliminary study. In this study the participants were asked closed-ended questions about all the factors which were stated previously in Section 3. The purpose of the questions was to measure the importance of the identified factors in the proposed model for cloud adoption from an expert perspective. The closed-ended questions were designed using a five-point

Likert scale, which ranged from 5 (very important) to 1 (not relevant). SPSS software was used to analyse the collected data from IT experts; the test value was identified as 3. Table 2 presents the results of using the one-sample t-test.

In this study Bonferroni correction was used for controlling for false positive results by dividing alpha (α) by the number of factors included in the questionnaire.

$$(\alpha/n) = 0.05/15 = 0.0033 \quad (1)$$

Table 2: One-sample t-test.

Factors	p-value	Result
Availability	<0.001	Statistically significant
Reliability	<0.001	Statistically significant
Security	<0.001	Statistically significant
Privacy	<0.001	Statistically significant
Trust	<0.001	Statistically significant
Relative advantage	<0.001	Statistically significant
Compatibility	<0.001	Statistically significant
Complexity	<0.001	Statistically significant
Top management support	<0.001	Statistically significant
Organisation size	.003	Statistically significant
Technology readiness	<0.001	Statistically significant
Compliance with regulations	<0.001	Statistically significant
Competitive pressure	.008	Not statistically significant
Trading partner pressure	.148	Not statistically significant
Physical location	<0.001	Statistically significant

It is interesting to note that most of organisations taking part in this preliminary study were concerned about privacy, security and trust issues and this was one of the major reasons behind their decisions not to use cloud services. Furthermore, there were other factors that were suggested by experts, such as compatibility, compliance with regulations and cost savings, and organisations need to take these into account before employing the cloud services. Most of these factors already exist in the proposed model for cloud adoption.

In order to measure the reliability of the results, Cronbach's alpha was used in this initial study. According to Hinton (2004) and Field (2009), a value from 0.9 and above is considered highly reliable and from 0.7 to 0.8 is acceptable. The Cronbach's alpha coefficient of this study was 0.719, which is considered to be an acceptable value.

6 CONCLUSIONS

The great benefit of cloud technology is that the cloud offers resources to multiple users at any time in a dynamic way and according to user needs. In addition, users only pay for the services that they consume. However, despite the fact that the cloud offers various benefits for enterprises, from flexibility to cost reduction, moving data from an in-house data centre to the cloud is not a simple task. Therefore, this study seeks ways to encourage organisations to adopt cloud services in Saudi Arabia as well as to investigate the factors that affect the implementation of this technology. This paper presents the initial model for cloud adoption in Saudi Arabia and in future a survey will be conducted to validate the developed model. Further outcomes will be published shortly.

REFERENCES

Alkhater, N., Wills, G. & Walters, R. 2014. Factors influencing an organisation's intention to adopt cloud computing in Saudi Arabia. *IEEE 6thInternational Conference on Cloud Computing Technology and Science,* pp. 1040–1044.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., et al. 2010. A view of cloud computing. *Communications of the ACM,* 53, pp. 50–58.

Borgman, H. P., Bahli, B., Heier, H., & Schewski, F. 2013. Cloudrise: exploring cloud computing adoption and governance with the TOE Framework. *46thHawaii International Conference on System Sciences,* pp. 4425–4435. IEEE doi:10.1109/HICSS.2013.132.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. &Brandic, I.2009. Cloud computing and emerging IT platforms: Vision, hype and reality for delivering computing as the 5th utility. *Future Generation Computer Systems,* 25(6), pp. 599–616.

Buyya, R., Calheiros, R. N. & Li, X. 2012. Autonomic cloud computing: Open challenges and architectural elements. 2012 *Third International Conference on Emerging Applications of Information Technology,* pp. 3–10. doi:10.1109/EAIT.2012.6407847.

Chang, V., Walters, R. J., & Wills, G. (2013). The development that leads to the Cloud Computing Business Framework. *International Journal of Information Management,* 33(3),pp. 524-538.

Chang, V. (2015). A proposed Cloud Computing Business

Framework. ISBN: 9781634820172 (print), Nova publisher.

Chau, P.Y.K. and Tam, K.T. 1997. Factors affecting the adoption of open systems: An exploratory study.*MIS Quarterly,* 21(1), pp. 1–24.

Chen, X., Wills, G., Gilbert, L. and Bacigalupo, D. 2010. Using cloud for research: A technical review. JISC Final Report.

Field, A. 2009. *Discovering statistics using spss*. 3rded. Thousand Oaks, CA: Sage Publication.

Foster,I., Zhao, Y., Raicu, I.& Lu, S. 2008. Cloud computing and grid computing 360- degree compared. In: *Grid Computing Environments Workshop (GCE'08),* IEEE Press, pp. 1–10.

Hajjat, M., Sun, X., Sung, Y.W.E., Maltz, D., Rao, S., Sripanidkulchai, K. & Tawarmalani, M. 2010. Cloudward bound: Planning for beneficial migration of enterprise applications to the cloud. *ACM SIGCOMM Computer Communication Review,* 40(4), pp. 243–254.

Hao, W., Yen, I.-L. & Thuraisingham, B. 2009. Dynamic service and data migration in the cloud. *33rdAnnual IEEE International Computer Software and Applications Conference,* pp. 134–139. doi:10.1109/COMPSAC.2009.127.

Hinton, P. 2004. *Statistics Explained: A Guide for Social Science Students*. 2nded. Taylor & Francis.

Hu, J. & Klein, A. 2009. A benchmark of transparent data encryption for migration of web applications in the cloud. 2009. *Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing,* pp. 735–740. doi:10.1109/DASC.2009.85.

Jeffery, K. and Neidecker-Lutz, B. 2010. *The future of cloud computing opportunities for European cloud computing beyond.* Expert Group Report, Public Version 1.0.

Kaisler, S. and Money, W. H. 2011. Service migration in a cloud architecture. *44thHawaii International Conference on System Sciences,* pp. 1–10. doi:10.1109/HICSS.2011.371.

Khajeh-Hosseini, A., Greenwood, D. &Sommerville, I. 2010. Cloud migration: A case study of migrating an enterprise IT system to IaaS. *IEEE 3rdInternational Conference on Cloud Computing,* Cloud 2010, pp. 5–10, July 2010: Miami, FL, USA.

Khajeh-Hosseini, A., Sommerville, I., Bogaerts, J. & Teregowda, P. 2011. Decision support tools for cloud migration in the enterprise. *IEEE 4thInternational Conference on Cloud Computing,* pp. 541–548, (Khajeh-Hosseini et al. 2011a).

Khajeh-Hosseini, A., Greenwood, D., Smith, J. W. &Sommerville, I. 2011. The cloud adoption toolkit: Supporting cloud adoption decisions in the enterprise. *Software: Practice and Experience,* 42(4), (Khajeh-Hosseini et al. 2011b).

Klems, M., Nimis, J. & Tai, S. 2009. Do clouds compute? A framework for estimating the value of cloud computing. *Designing E-Business Systems.* Markets, Services and Networks, pp. 110–123. Springer Berlin Heidelberg.

Low, C., Chen, Y. & Wu, M. 2011. Understanding the determinants of cloud computing adoption. *Industrial Management & Data Systems,* (111)7, pp. 1006–1023. doi:10.1108/02635571111161262.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. & Ghalsasi, A. 2011. Cloud computing—The business perspective. *Decision Support Systems,* (51)1, pp. 176–189. doi:10.1016/j.dss.2010.12.006.

Rogers, E. M. 1995. *Diffusion of innovation.* 4thed. New York, NY: The Free Press.

Scott, W.R. and Christensen, S. 1995. *The institutional construction of organizations: International and longitudinal studies.* Thousand Oaks, CA: Sage.

Scott, W.R. 2001. *Institutions and organizations.* 2nded. Thousand Oaks, CA:Sage.

Tessmer, M. 1993. *Planning and conducting formative evaluations.* London: Kogan Page.

Tornatzky, L.G. and Fleischer, M. 1990. *The process of technological innovation.* Lexington, MA: Lexington Books.

Migration of Cloud Services and Deliveries to Higher Education

Raed Alsufyani, Fash Safdari and Victor Chang
School of Computing, Creative Technologies and Engineering, Leeds Beckett University, Headingley, Leeds LS6 3QR, U.K.
r.alsufyani5478@students.leedsbeckett.ac.uk, {f.safdari, v.i.chang}@leedsbeckett.ac.uk

Keywords: Cloud Computing, Cloud Computing Business Framework, Quality of Service, Quality of Experience.

Abstract: This paper discusses the adoption of cloud computing in education. It emphasizes the view that cloud computing is vital in the education sector because of its ability to reduce the overall costs of IT infrastructure installation and maintenance, improvement of efficiency, and the sharing of IT resources among students. The flexibility of cloud computing and its reliability makes it more appropriate for use in the educational environment. The Leeds Beckett University cloud project utilizes the SAS Educational Value-Added Assessment System, which gives lecturers the opportunity to deliver accurate content to students while monitoring their progress. Contemporary educational institutions must look forward to improve their research and education through cloud computing.

1 INTRODUCTION

The emergence of cloud computing and its application to diverse fields such as education has brought about a lot of opportunities for improving efficiency of service provision (Sultan, 2010). The key categories of cloud computing that institutions could adopt include the public cloud, the private cloud, the hybrid cloud, and the community cloud. Educational institutions, including colleges and universities have been quick to adopt cloud computing to boost efficiency, minimize IT costs, and improve their research and academic processes. For instance, Kurelović et al., (2013) estimates that cloud computing in K-12 students could consume up to 35% of the IT budget in the coming few years. This is an indication of the expanding cloud computing services in education. Ercan (2010) agrees that the security, reliability, and economic nature of cloud computing play a vital role in the challenging environment of education where large volumes of data are stored. There are a few frameworks and amongst them, the Cloud Computing Business Framework (CCBF) (Chang et al., 2013a) has been regarded as a recommended cloud adoption framework because of its ability to classify business models, portability, organizational sustainability, and the linkage of service models. The SAS Educational Value-Added Assessment System stands out as one of the best for the educational sector, as evidenced from its use at Leeds Beckett University.

This paper explicates cloud computing to highlight its meaning, classifications, reasons for university adoption, frameworks for cloud computing, and the factors for deploying cloud computing in education.

1.1 Defining Cloud Computing

There is no standard definition of cloud computing as many IT professionals have come up with their own definitions. However, the commonly used definition indicates that cloud computing is a cluster of distributed computers that offer on-demand resources and services over a networked medium commonly the internet (Sultan, 2010). It is worth understanding that it entails the deployment of groups of remote servers and software networks, which allow the centralized storage of data and access to computer services through the internet (Mokhtar et al., 2013).

1.2 Classification of Cloud Computing

Cloud computing is clearly classified into four significant categories. The first category is the public cloud. According to Chang et al., (2013), this is where the entire computing infrastructure is located in the cloud provider's premises and the user has no physical control over it. A public cloud tends to use shared resources and might be vulnerable to

attacks.

The private cloud comes in as the second category. This entails one particular organization using the cloud infrastructure for its different operations. It is remotely located and is not shared with other organizations. The advantage of the private cloud is that the customer has control over the infrastructure, as it could be hosted internally or externally (Chen et al., 2014).

The third category is the hybrid cloud, which implies utilizing both the private and the public cloud depending on the purposes they serve. For instance, an organization could use the public cloud in activities such as customer interaction while securing its network using the private cloud.

The last significant category is the community cloud that entails the sharing of infrastructure between organizations with shared data, and other data management concerns. The advantage is that it could be hosted internally or externally depending on the institution's choice (Singhal et al., 2013). This would be the most relevant cloud for the academic community because it significantly minimizes costs through a cost-sharing approach. **The operational costs are significantly reduced because the cloud is shared across community members. The aspect of cost minimization is also seen in terms of augmenting existing data resources rather than building new internal environments (Youssef, 2012). Moreover, it makes it easier for educational institutions to administer cloud and the traditional data centre environments remotely hence cutting down overall costs of operation**. Again, it allows for control of the infrastructure by the institutions utilizing it (Chang et al., 2013a). **Internal control of the cloud facilitates real-time reporting and ordering through customizable management portal**. **Thirdly, the community cloud is relevant to educational institutions because of their effective security, privacy, and compliance. It is usually tailored in such a way that it can address unique security problems and regulatory needs relating to the institution (**Singhal et al., 2013**).**

In line with these categories of clouds, institutions could enjoy various service models. The first is the Infrastructure as a Service (IaaS). Almorsy et al., (2011) affirm that this category offers relevant products such as remote delivery through the internet of the entire computer infrastructure. For instance, it offers storage, virtual computers, and servers.

The second category is the Platform as a Service (PaaS). Jula et al., (2014) indicates that this service model has transformed the traditional delivery of computing services. For instance, the presence of this category has enabled cloud providers to remotely offer diverse products including the hardware, middleware, a database and the operating system (Singhal et al., 2013).

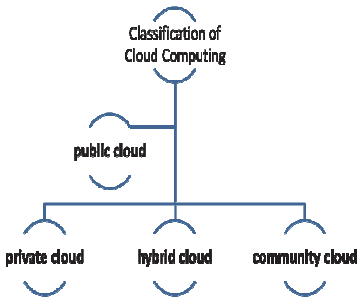

Figure 1: Classification of Cloud Computing.

The third model is Software as a Service (SaaS). This category delivers applications through the medium of the internet as a service. Users do not need to install and maintain software, as they have the pleasure of accessing it through the internet (Youssef, 2012). Overall, SaaS offers a complete application functionality stretching from productivity applications to other programs such as the Customer Relationship Management.

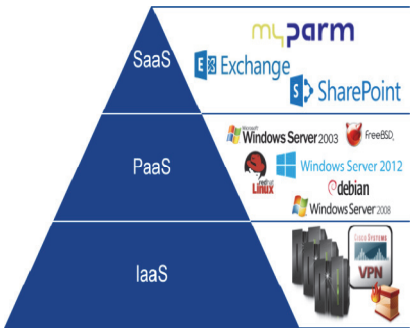

Figure 2: The Cloud-Computing-Architecture specified by the National Institute of Standards and Technology (NIST) knows three models, namely Infrastructure (IaaS), Platform- (PaaS) and Application models (SaaS).

1.3 Challenges of Cloud Computing

Whilst there are potential benefits associated with public cloud computing implementation, there are also risks and uncertainties that come with public cloud computing. Moving computing resources to the cloud is not without difficulties and issues. For example, Amazon cloud services outages caused many organizations the loss of their computing resources, services and incurred economical losses

(Bright, 2011). The potential benefits of cloud computing need to be assessed against possible associated risks.

Geczy et al., (2013) stated performance as one of concerns associated with cloud computing. In cloud computing, computing services and resources which were typically installed, managed, and accessed within organizations premises are hosted in data centres and in the majority of cases are accessed over the Internet. Internet is a best-effort and shared communication infrastructure. The organization data traffic has to travel through many different routes and hops shared by other organizations and user traffic, packets could travel over different routes which could be saturated and arrived out of sequence, packets could be lost (Ukil et al., 2013).

These could lead to delay and jitter resulting in poor performance. Reliability of Internet also plays a major role in the reliability and availability of the public cloud services. The internet is an unpredictable network environment (Ivanus and Iovan, 2014). These inherent characteristics of internet could have a major impact on the performance of public cloud. This paper aims at surveying the applicability of cloud technology in education. It is focused on encouraging all educational institutions to embrace cloud technology because of its advantages.

2 REASONS FOR UNIVERSITY CLOUD ADOPTION

Many universities around the globe have moved swiftly to incorporate cloud computing into their learning and research processes. Specifically, Chang and Wills (2013) inform that cloud computing is viewed as an attractive part of research and education within universities because of its ability to allow searches and collaborative working among students. **As previously discussed in section 1.2, the community cloud is the most relevant for universities because of its efficiency and cost reduction, security and privacy, and the agility in terms of service delivery (Chang et al., 2013a; Chang and Wills, 2013).** The University of Greenwich is one of the key institutions that have adopted cloud computing into their research and learning processes. In tandem with the University of Greenwich case study, five significant reasons have motivated universities to adopt cloud computing.

The first key reason for university cloud adoption is the fact that it plays an instrumental role in the reduction of environmental and financial costs in areas where services are needed for shorter periods (Chang et al., 2013a). This is bound to save many universities money, hence avoiding unnecessary wastage. Every university looks forward to cut down its operating costs in respect to IT and energy usage at the institution.

Secondly, many universities are adopting cloud computing because it has the capacity to make experiments more repeatable. Ercan (2010) points out with cloud computing in place, write-ups of science experiments conducted in the cloud could contain relevant applications such as the virtual machine that make them easy to replicate (Ukil et al., 2013).

Thirdly, Cloud adoption in universities is motivated by the understanding that it facilitates the sharing of the workload in cases where the university is working with another organization (Avram, 2014).

Another reason for cloud adoption in universities is that it allows and simplifies the access to web applications, data centres, and service from any given location within the university. Chen et al., (2014) agrees that this makes it easier for students to engage in research without having to concentrate in a particular area.

Lastly, universities are swiftly adopting cloud computing because of its flexibility and the aspect of pay as you go. There is always room to use specialized web-based software that otherwise could not be supported by in-house policies. This enhances the level of flexibility because of reduced bureaucracies among researchers (Ivanus and Iovan, 2014).

Apart from these motivations, the most significant challenge affecting cloud computing adoption and implementation is the lack of standards. Many institutions have found it difficult to define the desired standards. There have been no concerted efforts toward the definition of desirable standards relating to technological, management, and regulatory standards related to cloud computing (Avram, 2014)**. This poses the risk of failure in terms of take off and subsequent utilization of cloud computing in school settings.**

3 FRAMEWORKS OF CLOUD COMPUTING

One of the most effective and efficient framework that different academic institutions could adopt is the Cloud Computing Business Framework (CCBF).

The Cloud Computing Business Framework would be appropriate for the academic community because it plays an instrumental role in promoting a good cloud design **hence ensuring it works efficiently within the institution through the choice of a better pattern** (Chang et al., 2013a). **It also promotes deployment where all activities in the software system are assured through interrelated activities. The migration to the cloud and service models are also clearly assured through the CCBF framework** (Chang et al., 2013a). Specifically, the CCBF has four key areas that make it relevant to the academic community.

The first area is classification. This entails the categorization of diverse business models to offer cloud-adopting organizations relevant strategies and business cases. For instance, the educational community would be allocated its own strategy depending on its courses and other services (Viswanath et al., 2012).

The second relevant area that makes it appropriate is organizational sustainability. According to Vakil et al., (2013), it entails a structured framework that reviews the performance of the institution accurately. Every educational institution would want to operate at the best and most accurate level with the CCBF framework.

The fourth area that makes it relevant to the academic community is portability. Chang et al., (2013a) points out that they would be in a better position to manage the portability of applications to the cloud. With such portability, the academic community would also be in a better position to transfer applications between clouds offered by different vendors.

Lastly, the CCBF is appropriate for academic institutions because of its ability to link various cloud search approaches and service models such as the IaaS, PaaS, SaaS and the Business Models (Borgman et al., 2013). Overall, CCBF is justifiable for use in the academic community because of its simulations and ability to address every area.

Apart from this framework, the High Performance Computing (HPC) Framework could also be relevant for educational institutions. Its high computing capabilities and the ability to facilitate research among students puts it at a strategic position in terms of applicability to educational institutions. Again, it gives students the opportunity to access a shared pool of configurable computing resources including servers, networks, storage, and applications.

4 CLOUD IN EDUCATION: LITERATURE REVIEW

Wu (2010) points out that the adoption of the cloud into the field of education has been massive, as many educational institutions have taken the opportunity to maximize on its many advantages. The high level of cloud incorporation into the educational field emanates from its potential to improve efficiency, costs, and to improve convenience in the educational sector. Vakil et al., (2013) reiterates that numerous educational and official establishments in the U.S have continued to recognize the potential of cloud computing in terms of cost reduction and efficiency. The specific reasons that have motivated many educational institutions to adopt cloud computing include:

- The minimization of costs used in the IT infrastructure
- Attainment of efficiency in education delivery
- Improvement of convenience through features such as Pay-per-use
- Enhancement of resource consolidation
- Attainment of green IT
- In light of the above reasons, many universities have gone ahead to adopt cloud computing. For instance, the University of California at Berkley found out the significance of cloud computing in one of their courses that was solely focused on the development and deployment of SaaS applications (Fox, 2009). Donations from the Amazon Web Services (AWS) played an assistive role in helping the university move the course from a locally owned infrastructure to the cloud. It was noted that this would have enabled it acquires a large number of servers within the shortest time possible. This was also an opportunity to enhance resource consolidation at the university.

Economic reasons have pushed some educational institutions to adopt and utilize cloud computing in their learning environments. As noted earlier, cloud computing reduces costs significantly because it eliminates costs related to the development and maintenance of massive IT infrastructure (Jang, 2014).

- In line with economic conditions and the need to minimize costs, institutions such as the Washington State University's School of Electrical Engineering and Computer Sciences (EECS) have been forced to embrace cloud computing to cut down their operational costs.

They were able to select the vSphere4 platform, which is flexible, dynamic, reliable, and offers seamless maintenance of the IT infrastructure (Fox, 2009). Learning and research has been simplified through the cost rationalization approach, which recognizes the need to do more with less.

Cloud computing in education has been applied internationally in numerous educational institutions starting from primary schools to universities. Schools in European countries such as Britain have adopted the cloud in their educational system hence enhancing efficiency.

- Hicks (2009) affirms that some of the common examples of U.K universities that have been able to incorporate cloud computing into their academia include the Leeds Metropolitan University and the University of Westminster. The key factors for the move to the cloud was cost reduction and the enhancement of reliability in the use of computing services (Shin et al., 2014). They have also based three reasons on the need to enhance green IT in their learning environment. This has led to proper functioning and flexible operations in terms of research and academics.

  African educational institutions have not been left behind in terms of using the cloud in their research and learning. The lack of an adequate IT infrastructure and the inability to cope with software and hardware upgrades have contributed to the adoption of the cloud in many African educational establishments (Truong et al., 2012).
- With the help of Google, institutions such as the University of Nairobi in Kenya and the National University of Rwanda have embraced cloud computing. This has enhanced information sharing among students and has been critical to the minimization of costs related to IT maintenance, enhancement of flexibility, and resource consolidation. Microsoft is also helping Ethiopia roll out the project of distributing 250,000 laptops all operating on Microsoft's Azure Cloud platform (Sultan, 2010).

Therefore, the critical benefits of using cloud computing in education could be summarized as below.

1. Lower capital costs for institutions. Erkoç and Kert (2012) reiterate that this is especially because educational institutions have the opportunity to offer a wide variety of services while only paying for the actual capacity paid
2. It leads to flexibility in the provision of research and academic services because users can access it at any given location in the institution (Fox, 2009).
3. It saves on costs by over 50% related to the installation and maintenance of IT infrastructure in academic institutions
4. It offers an optimized and customizable IT infrastructure, which offers quick accesses to the desired computing services in the educational institution (Almorsy et al., 2011)

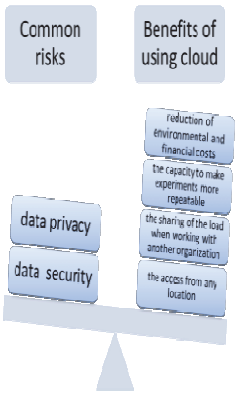

Figure 3: Deploying Cloud Computing in Education: benefits & common risks.

However, there are several risks associated with the use of cloud computing in the academic community. Common risks of using cloud computing in education include:

- The cloud is always subject to the risk of virtual exploits that target both the virtual host and its users. According to Almorsy et al. (2011), the common risks that could be suffered include guest-guest, guest-host, and host-guest virtual threats.

There is always the threat to data privacy and security. The interaction between the cloud provider and the institution poses a major risk to data security, especially if the matter has not been critically analyzed (Ercan, 2010).

5 CLOUD COMPUTING FOR EDUCATION: DEPLOYMENT SCENARIO

This section presents Cloud Computing for Education. Several factors need to be keenly considered before migration of cloud computing services for education. Katz et al., (2010) reveals

that one of the relevant factors for deployment is where the cloud services are to be hosted. The educational institution could choose either the public or private development approach depending on the availability of financial resources to host the cloud within its premises.

The second deployment factor is security. Educational institutions must figure out the kind of data that would be put into the cloud (Lakshminarayanan et al., 2012). Sensitive information such as the institution's financial information would require a higher level of security. Therefore, they must weigh up the security of the system before deployment.

The third factor for deploying cloud computing in education is customization capabilities. Different educational institutions have different approaches to the learning processes (Mircea and Andreescu, 2011). Others would want to customize their services and products to students. Therefore, it is vital to understand whether the available cloud computing services are customizable to meet local needs.

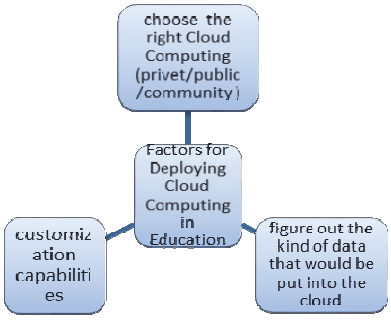

Figure 4: Factors for Deploying Cloud Computing in Education.

According to Powell (2010), the fourth factor is the legal requirements relating to the cloud. Educational institutions need to have an in depth understanding of the legal considerations and implications that might arise from security breaches in the cloud.

5.1 Quality of Service (QoS) and Quality of Experience (QoE) for Leeds Beckett SAS Cloud

The quality of service (QoS) is primarily used in monitoring the performance of the cloud service without necessarily reflecting the user's quality of experience. However, the quality of experience (QoE) makes up for this by considering the views of the person using cloud services for their activities (Safdari and Chang, 2014). In the educational sector, the improvement of cloud services could be realized through the quality of experience monitoring approach. **This is especially because it tends to focus on the entire service experience and it tends to offer a holistic evaluation of the system rather than focusing on narrowed experiences of users** (Safdari and Chang, 2014). The Leeds Beckett University cloud project is anchored on the ability of combining the cloud with big data techniques. For instance, the cloud project looks forward to the facilitation of matters such as Storage as a Service, Education as a Service, Business Intelligence as a Service, and Integration as a Service (Amrein-Beardsley and Collins, 2012). The Leeds Beckett University cloud project utilizes the SAS Educational Value-Added Assessment System (SAS EVAAS), which has been perceived as the most robust and reliable system available (Amrein-Beardsley and Collins, 2012). Accordingly, it uses SAS for several reasons. Firstly, SAS EVAAS helps lecturers at the university to measure the progress of students and accurately improve the delivery of their instruction. They are always able to modify the curriculum depending on the ability of their students to grasp the content being taught in class (Amrein-Beardsley and Collins, 2012). Additionally, it assists in the alignment of professional goals with the greatest needs of students and hence improving the ability of educators to deliver content. It basically helps in the evaluation of the effectiveness of educators in delivering their content to students. Secondly, policy makers at the university are able to conduct more rigorous longitudinal analysis of the student test results at the university with SAS EVAAS (Amrein-Beardsley and Collins, 2012). This is attained through the assessment of the accessibility of students to opportunities and services offered through the cloud. The educational environment would be more efficient with the use of the SAS in their cloud computing system because of the enhancement of teaching strategies and student success.

6 A CASE STUDY AT LEEDS BECKETT UNIVERSITY

Since a number of universities do not publish their their Educational Cloud projects publicly, there is a need to disseminate lessons learned and recommendations in the Higher Education. This section presents a case study for Leeds Beckett University's case study of our Cloud project,

including the current status, technologies and useful lessons learned.

6.1 Illustration of Leeds Beckett SAS Cloud

SAS Cloud has been used as a platform and language for business intelligence (BI) at Leeds Beckett University since Year 2012. The aim is to improve the quality of education and students' experience through the interactive platform provided by SAS. The objective is to develop a master's program in Business Intelligence, which includes modules such as "Business Intelligence, Data Analysis and Visualization" (BIDAV), "Data Warehouse", "Advanced Data Warehouse" and elective modules. Amongst all these modules, BIDAV is the one that provides students both theoretical foundations and practical learning experience, in which students have to learn the SAS programming and use it for developing BI code. BI is a popular topic, in which Chang (2014) has demonstrated how to design, implement and analyze a Business Intelligence in the Cloud to calculate risk and return for financial stock options. The ability to process, interpret and utilize data, as well as understand complex data analysis, is an important skill for employability. Similarly BI concepts can be fully transferable to Higher Education to ensure that students can equip with numeracy, quantitative and analytical skills required by employers. With more training in place, students can build up their competency and demonstrate their BI portfolios and services. They can import their datasets directly into SAS, which have built-in libraries and server connected directly to the Cloud in the US. Upon clicking "run", their code will be executed directly on the SAS Cloud in the US as shown in Figure 5.

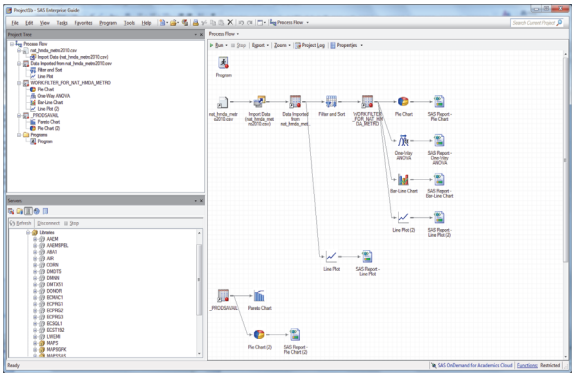

Figure 5: A screenshot about SAS Cloud.

Students can receive all their data analysis results, interpretation and visualization in one go with less than 10 seconds of waiting time. Table 1 shows an example for SAS syntax. SAS is a procedure-driven language, meaning that all the steps have been predefined. The emphasis is to show a list of useful procedures to students and explain how they can be used in different cases. In Table 1, "autoreg" is a procedure to perform regression which can generate statistical tests and data analysis simultaneously. All these take a matter of seconds for students. The data used for analysis is called "pred", which uses autoreg for computation and then calculates the residual (the sum of all the differences between all datapoints and regression line) from the statistical.

6.2 Data Visualization

Data Visualization is an important aspect in learning business intelligence, in which students can directly understand the interpretations of data and its correlations with other aspects of data analysis. Since there are several statistical results and tests, it is difficult for some students to understand the meanings of all these outputs (Chang, 2014). Hence, the use of data visualization is extremely useful for students to understand complex datasets and their correlations to other data.

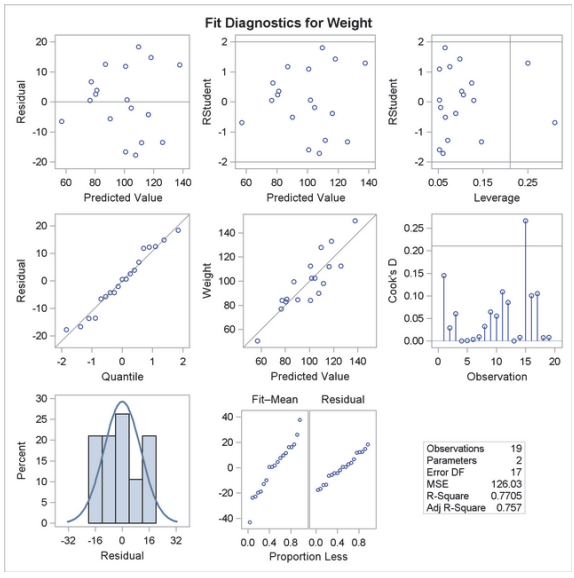

Figure 6: An example of data visualization with statistical tests and key outputs.

According to our experience, students can understand the module, BIDAV, much quicker than using traditional means of learning. In order to understand the benefits and long-term implications,

questionnaires will be designed to identify the improvement in learning efficiency and student satisfaction in our next phase of research. Figure 6 shows the screenshot of a data visualization output, where all the key results can be computed simultaneously along with different types of statistical tests.

6.3 Discussion

Overall, it is clear that cloud computing is directly applicable to the educational sector because of the significant role it plays in boosting learning and research processes among students. The community cloud is the most recommendable because of its ability to consider the costs dedicated to IT infrastructure within learning institutions. The sharing of resources plays an instrumental role in ensuring that costs are minimized in the best ways possible (Lakshminarayanan et al., 2012). The adoption of cloud compution at Leeds Beckett University provides a reliable case study for the best ways to adopt and implement cloud computing within educational institutions. The main aim of using SAS at the institution is to improve the educational outcomes of students through closer monitoring by lectures. Hence, significant efforts will be made for future research to explore the best cloud computing frameworks being instrumental in boosting educational outcomes in educational institutions (Viswanath et al., 2012). Furthermore, there should be an exploration of how costs could be minimized through resource sharing among educational institutions in their sharing of cloud services. This will make it easier for institutions to implement the project without being limited by their tight budgets or financial situation. Another approach is to use an implementation framework to ensure that all Cloud services can be delivered on time (Chang et al. 2013 b). The example include the integration between Education as a Service and Storage as illustrated by Chang et al., (2013 b).

7 CONCLUSION

In conclusion, cloud computing is geared toward transforming the educational field through efficient and reliable operations. Sharma and Ganpati (2013) conclude that the ability of cloud computing to reduce overall costs relating to IT infrastructure and its capacity to boost data access at any given location in educational institutions has been crucial in promoting its adoption and usage in the educational field. Universities such as the University of Greenwich have been on the forefront of adopting cloud computing and using it for research and educational services. Additionally, Leeds Beckett University's SAS Cloud system has worked more efficiently by giving lecturers the opportunity to accurately monitor the progress of its students in education and research. The educational field around the world is gradually becoming technological thanks to the evolution of cloud computing. The reliability and efficiency of cloud computing presents hope for the continuous growth of the educational sector. However, educational institutions need to beware of the common risks they could face while utilizing cloud computing, such as virtual threats and the loss of vital information. There needs to be clear communication with the cloud providers to mitigate such risks and benefit continuously from the cloud in terms of minimal capital costs, flexibility, and the provision of customized services to students.

REFERENCES

Almorsy M., Grundy, J. & Ibrahim, A. S., 2011, 'Collaboration-based cloud computing security management framework,' *IEEE International Conference on Cloud Computing,* pp. 1-8.

Amrein-Beardsley, A. & Collins, C., 2012, 'The SAS Education Value-Added Assessment System (SAS EVAAS) in the Houston Independent School District (HISD): Intended and Unintended Consequences,' *Educational Policy Archives*, vol. 20, no. 12, pp. 1-31.

Avram, M. G., 2014, 'Advantages and challenges of adopting cloud computing from an enterprise perspective,' *Procedia Technology,* Volume 12, pp. 529 – 534 .

Borgman, H. P., Bahli, B., Heier, H. & Schewski, F., 2013, 'Cloudrise: Exploring cloud computing adoption and governance with the TOE framework,' *46th Hawaii International Conference on System Sciences,* pp. 4425-4435.

Chang, V., Walters, R. & Willis, G. 2013a , 'The development that leads to the Cloud Computing Business Framework,' *International Journal of Information Management,* vol. 33, no. 3, pp. 524-538.

Chang, V., Walters, R. J., & Wills, G. 2013 b. 'Cloud Storage and Bioinformatics in a private cloud deployment: Lessons for Data Intensive research'. In *Cloud Computing and Services Science* (pp. 245-264). Springer International Publishing.

Chang, V. & Wills, G., 2013, 'A University of Greenwich case study of cloud computing: Education as a service,' *IGI Global Disseminator of Knowledge,* vol. 1, no. 1, pp. 1-22.

Chang, V., 2014. 'The Business Intelligence in the Cloud',

Future Generation Computer Systems, 37, 512-534.
Chen, S. L., Chen, Y. Y. & Hsu, C., 2014, 'A new approach to integrate internet-of-things and software-as-a-service model for logistic systems: A case study,' *Sensors,* vol. 14, no. 4, pp. 6144-6164.
Bright, P., 2011, 'Amazon's lengthy cloud outage shows the danger of complexity', [Internet], Available from: <http://arstechnica.com/business/2011/04/amazons-legthy-outage-shows-the-danger-of-complexity/> [Accessed 15-September-2013].
Ercan, T., 2010, 'Effective use of cloud computing in educational institutions,' *Procedia Social and Behavioral Sciences,* vol. 2, no. 1, pp. 938–942.
Erkoç, M. F. & Kert, S. B., 2012, 'Cloud computing for distributed university campus: A prototype,' *International Conference The Future of Education,* pp. 1-4.
Fox, A., 2009, '*Cloud computing in education,'* Viewed 24 February 2015, <https://inews.berkeley.edu/articles/Spring2009/cloud-computing>
Géczy, P., Izumi, K. and Hasida, K., (2013), 'Hybrid Cloud Management: Foundation and Strategies', National Institute of Advanced Industrial Science and Technology.
Koetsier, J. (2013). '10 million Malaysian students, teacher, and parent will now use Goggle Apps for Education', [Internet], Available from: <http://venturebeat.com> [Accessed 12-June-2013].
Hicks, B 2009. UK universities put their faith in the Google cloud*,'* viewed 24 February 2015, <http://www.agent4change.net/resources/open-source/280-uk-universities-put-their-faith-in-the-google-cloud.html>
Ivanus, C. & Iovan, S., 2014, 'Cloud computing technology trends,' *Fiability & Durability / Fiabilitate Si Durabilitate,* Volume 1, pp. 264-269.
Jang, S., 2014, 'Study on service models of digital textbooks in cloud computing environment for SMART education,' *International Journal of u- and e-Service,* vol. 7, no. 1, pp. 73-82.
Jula, A., Sundararajan, E & Othman, Z 2014, 'Cloud computing service composition: A systematic literature review,' *Expert Systems with Applications,* vol. 41, no. 8, p. 3809–3824.
Katz, R., Goldstein, P. & Yanosky, R., 2010, 'Cloud computing in higher education,' *Article,* pp. 1-12.
Kurelović, K, Rako, S & Tomljanović, J 2013, 'Cloud computing in education and student needs,' *MIPRO,* vol. 2, no. 2, pp. 856-861.
Lakshminarayanan, R., Kumar, B. & Raju, M., 2012, 'Cloud computing benefits for educational institutions,' *Higher College of Technology,* pp. 1-7.
Mircea, M. & Andreescu, A. I., 2011, 'Using cloud computing in higher education: A strategy to improve agility in the current financial crisis,' *IBIMA Publishing ,* Volume 2, pp. 1-15.
Mokhtar, S. A., et al 2013, 'Cloud computing in academic institutions,' *ICUIMC,* pp. 1-7.
Powell, J., 2010, 'Cloud computing – what is it and what does it mean for education?,' *Leicester Business School,* pp. 1-8.
Safdari, F. & Chang, V., 2014, 'Review and analysis of cloud computing quality of service,' *School of Computing*, pp. 1-7.
Sharma, A. K. & Ganpati, A., 2013, 'Cloud computing: An economic solution to higher education,' *International Journal of Application or Innovation in Engineering & Management (IJAIEM),* vol. 2, no. 3, pp. 200-206.
Shin, J., Jo, M., Lee, J. & Lee, D., 2014, 'Strategic management of cloud computing services: Focusing on consumer adoption behavior,' *Engineering Management, IEEE Transactions,* vol. 61, no. 3, pp. 419 - 427.
Singhal, M. et al., 2013, 'Collaboration in multicloud computing environments: Framework and security issues,' *IEE Transactions on Cloud Computing,* vol. 46, no. 2, pp. 75-84.
Sultan, N., 2010, 'Cloud computing for education: A new dawn?,' *International Journal of Information Management,* vol. 30, no. 1, pp. 109–116.
Truong, H. L., Pham, T. V., Thoai, N. & Dustdar, S., 2012, 'Cloud computing for education and research in developing countries, ' *IGI Globa,* pp. 78-94.
Ukil, A., Jana, D. & Sarkar, A. D., 2013, 'A security framework in cloud computing infrastructure,' *International Journal of Network Security & Its Application,* vol. 5, no. 5, pp. 11-24.
Vakil, F., Lu, V. & Russkoff, A., 2013, 'Recent developments in cloud computing and high speed connections for business practices,' *Review of Business,* vol. 33, no. 1, pp. 111.
Viswanath, D., Kusuma, S. & Gupta, S. K., 2012, 'Cloud computing issues and benefits modern education,' *Global Journal of Computer Science and Technology Cloud & Distributed,* vol. 12, no. 10, pp. 1-7.
Wu, C. F., 2010, 'Impact on applying cloud computing service to IT education,' *Department of Information Management,* Volume 168, pp. 170-175.
Yang, H. & Tate, M., 2012, 'A descriptive literature review and classification of cloud computing research,' *Communications of the Association for Information Systems,* Volume 31, pp. 35-60.
Youssef, A. E., 2012, 'Exploring cloud computing services and applications, '*Journal of Emerging Trends in Computing and Information Sciences,* vol. 3, no. 6, pp. 838-848.

Design of Smart Business-oriented Mining Engine

Neil Y. Yen and Jason C. Hung
School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu, Japan
Department of Information Technology, Overseas Chinese University, Taichung, Taiwan
neilyyen@u-aizu.ac.jp, jhungc.hung@gmail.com

Keywords: Data Fusion, Multi0-layered Fusion, Mining Engine, Planning-based Prediction, Smart Business.

Abstract: Keys to successful implementation of smart business require a wide spectrum of domain knowledge, experts, and their correlated experiences. Excluding those external factors – which can be collected by well-deployed sensors – being aware of user (or consumer) has the highest priority on the to-do-list. The more user is understood, the more user can be satisfied from an intuitive point of view, and thus, data plays a rather essential role in the scenario. However, it is never easy to achieve comprehensive understanding as the data requires further processing before its values can be extracted and used. So how the data can be properly transformed into something useful for smart business development is exactly what we pursue in this study. As a pioneer, three major tasks are focused. First, a mining engine is developed to be responsible for the universal collection of data which is primarily from real world, cyber world, and social world. Second, we go further into the fusion process of the collected data (e.g., the consumer purchase data shared by real-world company). A three-layer analysis and mining procedure is designed to enhance the mining engine through conventional RFM (Regency, Frequency, and Monetary Value) model and a set of fusion techniques. And in the end, we make planning-based predictions for a real-world company for expansion of the business interests.

1 INTRODUCTION

Smart business, by definition, indicates the ability to achieve goals which are set according to the development tendency of business (Watson et al., 2007). The key to successful implementation of the vision of smart business relies on a comprehensive understanding to the surrounded scenario in which wide spectrum of elements are concerned. Instances simply include vision of company, global economics situation, moving trends, targeted market and consumers, and etc. It is never difficult to find thousands of similar elements for consideration. But however, all these elements are useless unless they are well collected in form of data for further analysis (Cody et al, 2002).

Transforming data into meaningful and useful information (Parsons, 1996) that support the implementation of smart business is a long journey. Although rapid development in information communication technology makes it easy for data retrieval nowadays, sources where the data may be retrieved vary. The technology has also brought a tremendous change on our living world world – Hyper World (Kunii et al., 1996) – in which data is supposed to be from diverse channels and in unstructured formats. As such, how the data is retrieved, managed and processed becomes an open challenge when the issue concerning the comprehensive understanding is mentioned.

Collecting data, as much and complete as possible, is the first step to ensure enough and necessary information can be obtained. But this is, however, never taken as a practical way since decisions are made momentarily and sometimes only with limited information input. And thus, choosing one aspect as an entry point is a feasible action in the whole scenario. The end user, in general, is then considered a direct and intuitive way for this purpose.

Understanding the needs and preferences of users becomes complicated and requires more efforts than it used to be since users are spending more and more time on their activities like on-line shopping, interactions, communications, etc. in the Cyber world (Ma et al, 2011) via social media rather than face-to-face in the Real world. One of indispensable efforts is to collect their activity data in Hyper

World, mine their features, and discover their needs and preferences. This is a normal trilogy in the big data era. Data mining engine in this trilogy is essential for big data mining. In recent years, it has received great amount of attentions from academic society, industry, and business corporations. In particular, Google in 2011 released Data Mining Engine called Correlate, which enables users to find matching search trends. Oracle Data Mining Engine (DME) is the infrastructure that offers a set of in-database data mining functionality to its JDM (Java Data Ming) clients via a DME connection object. Amazon, the retail giant has been focusing on product recommendation engine, but recently, released Amazon Kinesis (Varia et al, 2014), which is streaming data real-time processing engine. It looks like almost data mining systems provide suites of data mining tools or software and put efforts on dealing with big and streaming data but how to efficiently meet application requirements and associated design approaches are not clearly mentioned or described.

Following the above-mentioned challenges and from the perspective of maximizing the benefits of business, this research pays the emphasis on the design of a universal framework for smart business support. This framework is instanced by a set of fusion techniques and a mining engine, and outcomes the planning-based predictions for a local company inside Japan. This smart business framework targets to provide services that best meet the needs of end users, retain the loyalty of existing users, and attract new users.

Meanwhile, descriptions to the proposed fusion techniques, data-data (D-D) fusion, algorithm-algorithm (A-A) fusion, feature-feature (F-F) fusion, data and algorithm (D-A) fusion, data and feature (D-F) fusion, and algorithm-feature (A-F) fusion, and data-algorithm-feature (D-A-F) fusion, will be elaborated. The input of the data mining engine is datasets and the output can be data, information, and knowledge, which are the input of Knowledge-Information-Data fusion engine or each of them can be used as a service (data as a service (DaaS), information as a service (IaaS), knowledge as a service (KaaS) directly to end user services.

Rest organization of this paper includes: Section 2 details the previous studies that relate to this study; Section 3 addresses the design of the fusion technique-based smart business framework; Section 4 gives a case study demonstrating the feasibility and preliminary results with the support of proposed framework; and Section 5 then concludes this paper and indicates potential extension of this work.

2 UNIVERSAL DESIGN OF FUSION TECHNIQUE-BASED SMART BUSINESS FRAMEWORK

Key to successful implementation of a smart business paradigm relies on many aspects. Excluding those that strongly require matured domain knowledge, the most common one is to satisfy the targeted audience, which is the user (or consumer as well), at the most. One significant instance is the service provision. A great amount of profits can be guaranteed if the service(s) to targeted consumers is right-to-the-needs.

A universal framework towards the implementation of smart business is then designed to this end. The proposed framework indicates an integrated approach, concerning the well transformation process from data to knowledge, together with a set of fusion techniques in interdisciplinary fields. A standard process that facilitates such process (i.e., diverse and unstructured data to well-defined information) is expected for future usage.

Figure 1 describes the image of our smart business framework. Five major portions are included:

(1) ***Data Acquisition*** is a universal entry for data collection. The data sources primarily contain the data in real world including weather information (e.g., temperature, atmosphere, quantity of rainfall, etc.), geographical information (e.g., coordinate, topography, etc.), human-related activities (e.g., supplies, equipment, manpower, etc.) that can be retrieved through deployed sensors; cyber data retrieved from social media such as a tweet/retweet from Twitter, a post (i.e., check-in, photo, message) on Facebook, an instant message via instant communication applications on smart devices; and other associated or related environmental data provided by third-party companies or organizations;

(2) ***Data Mining Engine*** is the fundamental component that connects the input data source and the follow-up data processes. It is composed by a set of mining techniques (e.g., statistics algorithms, practical machine learning tools) to meet all the necessary needs from users. It is responsible for the analysis of retrieved data, especially those multi-dimensional

data such as contextual, spatial, temporal, topical information with huge volume and high complexity, from available channels. Among all these methods, this engine is especially designed to incorporate possible fusion process, at the level of data, which may take place while specific requests are given by the users with heterogeneous data. Concerning the real-world situation, an integrated approach, i.e., the three-layer analysis and mining procedure, is proposed to cooperate with those existing, e.g., one-step, data fusion and mining algorithms. This approach, in particular, dynamically adjust the data for the fusion process, and select appropriate mining algorithms for execution;

(3) ***KID (Knowledge-Information-Data) Fusion Engine*** represents the second-step of fusion process in the framework. It especially concentrates on the fusion of processed input, which means, every stages of input, even knowledge, information, and raw data itself, may be fused in the case of necessary;

(4) ***Consumer Behavior Model*** contains a set of training and learning algorithms that continuously support the understanding of targeted users of a company. This model concentrates on the reuse of collected data correlated to users to shape the users and group them, depending on specific situations, as well. With several times of alternation, most explicit behaviors can be well predicted; and

(5) ***Open Platform*** is an universal portal that connects our proposed framework and external service, or data, providers. It enables the consumer behavior model to be built and grown, not only from the business point of view via the data mining engine and the data fusion engine but also from third-party contributions. It is designed to accept any trusted requests from the partners, and these accesses are also applied to enhance the proposed framework for better results provision.

A wide range of elements for the sake of better improvement in fusion techniques are considered while this framework is designed. This framework, and thus, identifies a general design to the whole scenario that take place in the implementation phrase of smart business. In other words, this framework is applicable to be further exploited to meet any specific purposes and cases.

In order to examine the feasibility, this paper especially concentrates on its usage to advance three essential issues, which are also the basics in the whole scenario, of smart business. The data mining engine of this framework is expected to lead preliminary solutions for a real-world retail company to:

a) **Find out the motivation of consumers and keep them connected:** The data, such as the personal information, preference, records of browsing and purchasing, activities on the Internet, device(s) used, and any possible activities on the social network (Moutinho, 1987), are collected and analyzed to provide better purchasing experience (i.e., personalized product and browsing on the website).

b) **Find out the elements that best attract consumers:** The above-mentioned data is further translated into information for self-training and learning processes. This information is expected to lead the element that creates the motivation of consumers. Monthly discount, free gifts, and jumping sales are taken as instance.

c) **Find out the thinking pattern of consumers:** For this purpose, the most efficient way is to allow seamless participation of consumers. No matter the comments or information shares over the social media or other related platforms by consumers shall be considered. With the trained information, the company may present new products that best meet the consumers, or a specific portion of them, to increase the business profits.

These three issues are taken as the primary concerns in data mining engine. Details of the design are introduced from the next section on.

3 DESIGN OF DATA MINING ENGINE

As stated above, this paper mainly focused on the design of data mining engine and its underlying fusion techniques for the planning-based product prediction for a real-world retail company though five core components are mentioned in the framework. As we know, collection of data with variety of types, huge volume, and high complexity

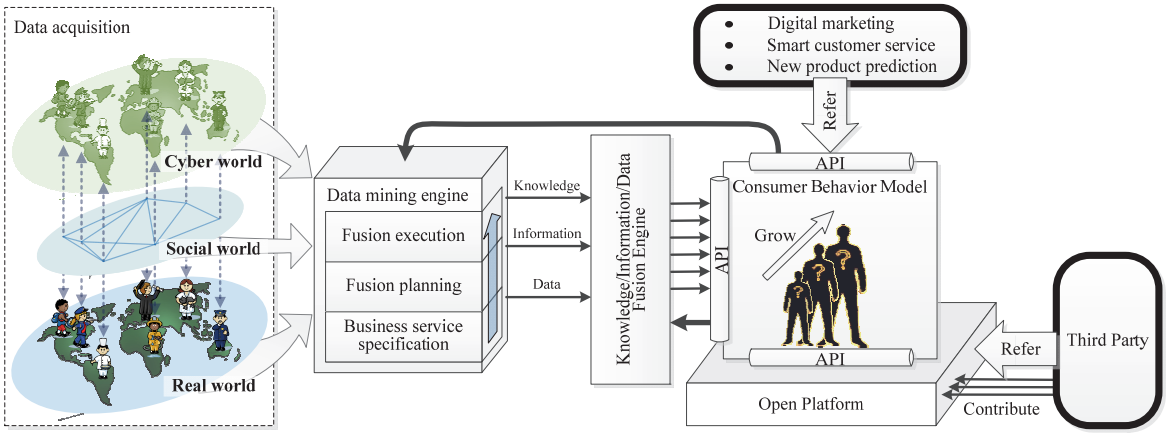

Figure 1: The Smart Business Framework.

is never easy and often requires corresponding hardware/software supports.

The fundamental concern to design a new type of data mining engine is that one single data mining method (or algorithm) or one-step mining procedure by conventional platforms or software tools are limited and not easy to meet all the possible needs in different phrases of service provision in smart business. Built-in fusion algorithm is important in the data mining engine. Our data mining engine, to this end, is featured by its dynamic mining and self-learning process with continuous input from the possible data sources. A three-level fusion technique for different business objectives and a set of fusion-based learning algorithms for prediction are developed.

3.1 Three-level Fusion Technique

The fusion techniques may be applied to three levels of fusion: data level (D-D fusion), algorithm level (A-A fusion), and feature level (F-F fusion). In addition, it may be necessary to merge the outcomes from three different level fusion, i.e., combinational D-A fusion, D-F fusion, A-F fusion, and D-A-F fusion. To a specific business objective or expected features from the data mining engine, an objective oriented fusion management agent is designated for fusion planning.

As shown in Figure 2, the data mining engine performs a 3-step process, i.e., the business service specification, the fusion planning, the fusion execution. It includes the following components: an objective-feature_label table, a dataset pool, an algorithm pool, feature_label discovery function, data_attribute discovery function, data_fusion function, and algorithm_fusion function.

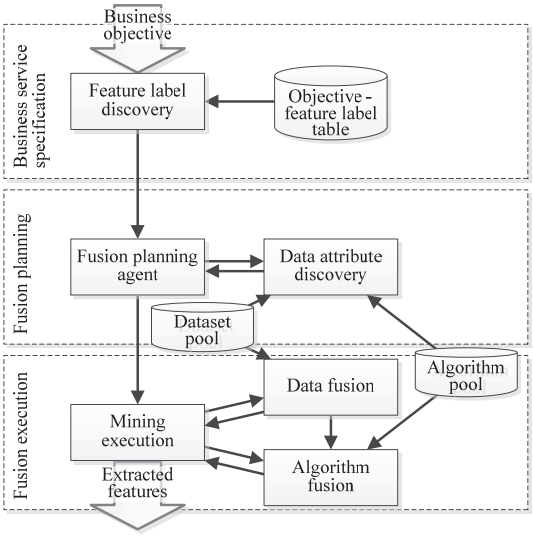

Figure 2: Process flow of fusion technique-based data mining engine design.

The data mining engine starts with a specified business objective, and retrieves its associated feature labels from the objective_label table. For a feature label or a set of feature labels, the fusion planning engine is to discover and schedule a sequence of algorithms in the algorithm pool triggered by the feature label(s). The data attribute discovery engine checks if the data attributes required by the triggered algorithms are contained in the dataset pool. If yes, data attributes are fused from different datasets as the input of the triggered algorithm. If not, unfound data attribute(s) are regarded as feature label(s), the above process is recursively repeated until all data attributes are directly found or indirectly created by applying for data fusion algorithm(s). Each triggered algorithm is

associated with its required data attributes as its input and the data attributes are associated with datasets in the dataset pool. As a result, a sequence of algorithms is retrieved and scheduled for execution.

The fusion planning agent is to trigger a sequence of algorithms in the algorithm-pool and discover their required data attributes are in datasets in the data-pool. The data attributes discovery is a backward chaining recursive procedure as given below. The resulting fusion plan implicitly indicates a suite of fusion algorithms (all or a part of 7-type fusion algorithms) triggered.

Algorithm 1 Backward Chaining Fusion algorithm

Inputs:
a set of feature labels: L
Outputs:
a set of data attributes: B
a plan as a collection of Algorithms: P

```
BCF()
  if L is empty ∨ recursive-depth = d then
    return
  else
    Get each feature label l in L
    Check each algorithm A in algorithm-pool
    if a(i) in datasets in data-pool then
      add a(i) → B
      add A → P
    else
      a(i) → L
      call BCF()
    end if
  end if
end
```

3.2 Fusion-based Learning Method

The fusion-based learning method is proposed to obtain correlations, especially the implicit patterns, between the preference of consumer and their periodical purchase. Most of the cases nowadays pay attentions on the discovery of user preference and further recommend potential products to the specific consumers so as to attract their interests. But this approach, however, faces to the user side. From the perspective of provider side (i.e., the company in this study), it is better to achieve comprehensive understanding of the cycle of purchase in order to present appropriate products to their consumers. For instance, a company will never sell heavy jacket in summer, or tank shirt in winter. What actions the company may take is to avoid unnecessary expenses (e.g., number of stocks, etc.) by prediction.

It is true that different approaches meet different kinds of consumer in running a business. For those low-royalty consumers, the recommending products may reach better profits than presenting periodical products there and waiting for consumers' notification to a company. But for those high-royalty consumers, a company may better stand a passive position with products that may attract the consumers to obtain better profits.

4 A CASE STUDY ON A RETAIL BUSINESS

4.1 Retail Data

The two types of retail data were collected through the online retail business and kept in the dataset pool. One is the customer profile which is composed of 23 attributes including customer ID, registration date and site, date of birth, region, gender, received mail magazines, reward points, etc. The number of registered customers is over 250,000. Another is the purchase record, which is composed of 38 attributes including order ID, item ID, color, size, order date and time, order site, customer ID, rough address, amount of purchase, etc. The number of records is

Figure 3: Retail Data at a glance.

over 450,000, about one year data from August 2011 to September 2012 as given in Figure 3.

4.2 Business Objectives

For any company, it is necessary to set their business goal in various levels from abstract to concrete business projects such as predicting, developing, adverting new products, attracting new customers, rewarding old customers, etc. Two business goals below are taken as case study in this research, however, due to the 4-page limitation, only case-1 is used to explain our fusion techniques based approach in this section.

Case-1: Awarding top customers: Objective feature label table lists top/best customers and associated customer value, customer segmentation and customer scoring as feature labels.

Case-2: Predict new product tendency: Objective feature label table lists new product tendency prediction and associated classification of the products in the top customer records as feature label.

4.3 Apply Backward Chaining Fusion Technique

Let us apply the backward chaining fusion to the case-1. As show in Figure 4, its associated two feature labels are top_instances and customer_value. Searching through the Algorithm pool, the algorithm, Extracting_top_instances() is triggered.

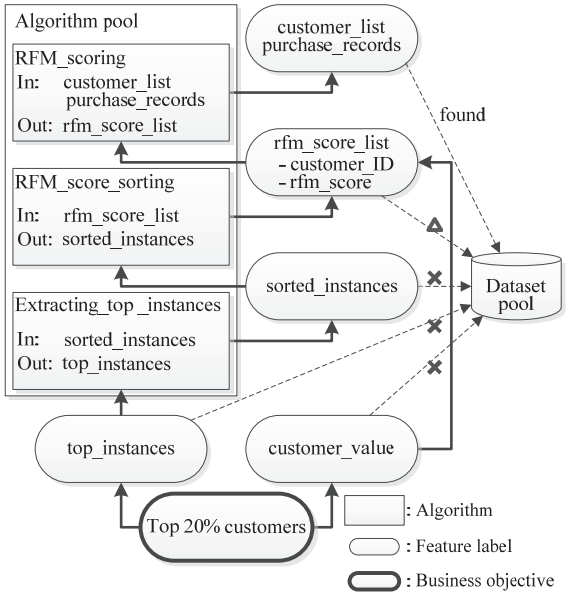

Figure 4: Applying BCF algorithm to case-1.

However, sorted_instances is not contained in the Dataset pool but it as a feature label triggers the algorithm, RFM_score_sorting() in the Algorithm pool. Again, rfm_sort_list including rfm_score are not in the Dataset pool but customer_ID can be retrieved from the Dataset pool while rfm_sort_list including rmf_score as a feature label triggers the algorithm, RFM_scoring(). Finally, required two attributes, customer_list and purchase_records are found in the Dataset pool. The backward chaining fusion algorithm terminated and a sequence of triggered algortihms, RFM_scoring(), RFM_score_sorting(), Extracting_top_instances(), with associated attributes, customer_list and purchase_records are the output of the fusion algorithm.

4.4 Analysis and Remarks

In implementing RFM_scoring(), it further requires R_scoring(), F_scoring(), and M_scoring() algorithms in the algorithm pool. These three algorithms can be called in parallel or sequence. Their results are fused, which is F-F fusion. Three algortihms, RFM_scoring(), RFM_score_sorting(), and Extracting_top_instances() are implemented in a sequence, which is A-A fusion. The customer_list and purchase_records are merged as RFM_scoring algorithm's input, which is D-D fusion. In other cases, other 4 types of fusion techniques may be applied.

5 CONCLUSIONS

This paper is mainly focused on our fusion technique based data mining engine which is the core component in the smart business framework. In this paper, 7-type fusion algorithms are listed, the fusion technique based data mining engine is described, the backward chaining based fusion planning engine as the heart of DME is explained. The case study on a practical retail business, a number of customer purchase record datasets is employed to show our design ideas and explain working principles of the proposed data mining engine.

Compared with other related work on data mining engine, our approach is the brand new in terms of building in 7-type fusion algorithms and having corresponding the backward chaining based fusion planning in the data mining engine.

ACKNOWLEDGEMENTS

If any, should be placed before the references section without numbering.

REFERENCES

T. L. Kunii, J. Ma and R. Huang, "Hyperworld Modeling", in proceedings of the *International Conference on Visual Information Systems*, pp1-8, Australia, February 1996.

J. Varia, S. Mathew, Overview of Amazon Web Services, http://media.amazonwebservices.com/AWS_Overview.pdf, January 2014, Amazon.

Jianhua Ma, Jie Wen, Runhe Huang, Benxiong Huang, "Cyber-Individual Meets Brain Informatics", IEEE Intelligent Systems , *Special Issue on Brain Informatics, Vol.26, No.5*, pp. 30-37, September/October 2011.

H. J. Watson, Barbara H. Wixom, "The Current State of Business Intelligence," Computer, 40(9), 96-99, 2007.

W. F. Cody, J.T. Kreulen, V. Krishna, W.S. Spangler, "*The Integration of business intelligence and knowledge management," IBM Systems Journal*, 41(4), 697-713, 2002.

N. Sun, J.G. Morris, J. Xu, X. Zhu, M. Xie, "iCARE: A framework for big data-based banking customer analytics," *IBM Journal of Research and Development, 58(5/6)*, 4:1-4:9, 2014.

S. Parsons, "Current approaches to handling imperfect information in data and knowledge bases," *IEEE Transactions on Knowledge and Data Engineering, 8(3),* 353-372, 1996.

L. Moutinho, "Consumer behaviour in tourism," *European journal of marketing, 21(10)*, 5-44, 1987.

Y. S. Wang, H.H. Lin, P. Luarn, "Predicting consumer intention to use mobile service," *Information Systems Journal, 16(2)*, 157-179, 2006.

J. C. Anderson, J.A. Narus, "Business marketing: understand what customers value*," Harvard business review, 76,* 53-67, 1998.

D. Boyd, K. Crawford, "Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon. Information," *Communication & Society, 15(5),* 662-679, 2012.

B. Xiao, I. Benbasat, "E-commerce product recommendation agents: use, characteristics, and impact," *Mis Quarterly, 31(1)*, 137-209, 2007.

D. F. Duhan, S.D. Johnson, J.B. Wilcox, G.D. Harrell, "Influences on consumer use of word-of-mouth recommendation sources," *Journal of the Academy of Marketing Science, 25(4)*, 283-295, 1997.

W. H. Delone, E.R. Mclean, "Measuring e-commerce success: Applying the DeLone & McLean information systems success model," *International Journal of Electronic Commerce, 9(1),* 31-47, 2004.

S. M. S. Hosseini, A. Maleki, M.R. Gholamian, "Cluster analysis using data mining approach to develop CRM methodology to assess the customer loyalty," *Expert Systems with Applications, 37(7*), 5259-5264, 2010.

R. Kohavi, L. Mason, R. Parekh, Z. Zheng, "Lessons and challenges from mining retail e-commerce data," *Machine Learning, 57(1-2),* 83-113, 2004.

H. U. Bauer, K.R. Pawelzik, "Quantifying the neighborhood preservation of self-organizing feature maps," *IEEE Transactions on Neural Networks, 3(4),* 570-579, 1992.

W. Hoeffding, "A class of statistics with asymptotically normal distribution," *The Annals of Mathematical Statistics, 293-325*, 1948.

An Overview of Cloud Services Adoption Challenges in Higher Education Institutions

Abdulrahman Alharthi, Fara Yahya, Robert J. Walters and Gary B. Wills
Electronics and Computer Science, University of Southampton, Southampton, U.K.
{aaa2g14, fby1g14, rjw1, gbw}@ecs.soton.ac.uk

Keywords: Cloud Services Adoption, Challenges, Cloud Computing, Higher Education, Integrated TAM Model.

Abstract: Information Technology (IT) plays an important role in enabling education services be delivered to users. Most education online services in universities have been run on the cloud to provide services to support students, lecturers, researchers and administration staff. These are enabled with the emergence of cloud computing in the world of IT. Cloud computing offers on demand Internet-based computing services. This paper presents an overview of cloud computing adoption in higher education, mainly tertiary institutions and universities. The focus of the paper is the challenges of cloud computing in higher education. It introduces the background to cloud computing and reviews research on adoption challenges in higher education institutions. These challenges are important as they provide an overview of the adoption of cloud in higher education. The authors proposed an integrated reference model based on the challenges in the literature integrated with TAM model to investigate the factors influence the users' attitudes and behaviours toward using cloud education services in universities ICT provision.

1 INTRODUCTION

In recent years, the Internet has accelerated the use of cloud services to support education online system. The cloud has become the main backbone in enabling such services by providing facilities to users. Cloud computing is known as a recent model that enables users to have computing resources on demand and pay per use (Sultan, 2010). It has been used widely in education; educators and students store and share their data widely in the cloud (Sultan, 2010). Previously, data were kept in external hard drives or storage servers in a location having restricted access in private networks. Nowadays, data can be stored in the cloud allowing accessibility to data to be more flexible and efficient.

Previous research has shown many aspects of cloud computing have been studied in the area of education, technology, education information systems (Alshwaier, 2012), integrating education resources and education system development (Huang, 2012). Smaller educational institutions often lack the resources or abilities to take full advantage of information technology. Cloud computing offers opportunities to improve the quality of education by offering flexibility and accessibility through the Internet. This can enable more dynamic and interactive learning experiences and allow students and teachers in multiple locations to collaborate and communicate more effectively (Alabbadi, 2011). In addition, cloud-based services can offer users and academic institution cost savings and access to scalable computing power (Buyya et al., n.d.; Armbrust et al., 2009; Motta et al., 2012).

2 BACKGROUND

This section presents the backgroud of cloud computing definitions, models and characteristics. Cloud computing computing is defined by the National Institute of Standards and Technology (NIST) as a model for providing a provisioned and on-demand computing resources which includes networks, servers, storage, applications, and services. It can be accessed using the Internet and needs minimal management effort or interaction from the cloud service provider (CSP) (Mell and Grance, 2011). Cloud computing is delivered at levels offering software applications, application platforms or various infrastructure elements as cloud systems. According to NIST (Mell and Grance, 2011), cloud computing has three service models:

- **Software as a Service (SaaS):** the entire system is cloud based, so users are presented with the application(s) only.
- **Platform as a Service (PaaS): s**uitable for user intending to deploy their own applications
- **Infrastructure as a Service (IaaS):** provides cloud based infrastructure such as storage, processing and networking elements.

Cloud computing is usually deployed in four models (Mell and Grance, 2011), Private Cloud, Community Cloud, Public Cloud, and Hybrid Cloud.

3 CLOUD IN HIGHER EDUCATION INSTITUTIONS

Higher Education institutions play an important role in the growth of societies. As with organisations nowadays, universities have become more reliant on Information and Communication Technology (ICT). ICT and internet-based services have to provide their stakeholders with educational services. Cloud computing is likely to be an attractive proposition to start up and small to medium educational establishments. The potential of cloud computing may include but is not limited to increasing service efficiency and cost-savings. An example from the University of California (UC) at Berkeley, found cloud computing to be attractive for use in one of their courses which was focused exclusively on developing and deploying SaaS applications (Alshwaier, 2012).

The Medical College of Wisconsin Biotechnology and Bioengineering Centre in Milwaukee found the use of cloud computing in their research has provided an astounding computing power. Researchers at the centre have been doing protein research which has been made more accessible to scientists from anywhere in the world. This is due largely to renting Google's cloud-based servers (Sultan, 2010).

Some universities have adopted cloud computing for economic reasons. The Washington State University's School of Electrical Engineering and Computer Science (EECS) has suffered cuts in its budget. However, the EECS claims that despite the challenging economic climate, cloud computing has actually enabled it to expand the services it offers to faculties and students (Sultan, 2010).

Some Universities are facing difficulties to provide scalable and flexible IT services. For instance, in traditional computer labs, there are many challenges present such as, limitation of lab hours and seats during the peak hours, repairing and maintaining computer labs, traveling to and from university, cost of outfitting traditional computer lab (hardware and software). Normally, IT services required by students, researchers and academic are requested from the IT Department, whose job is illustrated in Figure 1.

The IT department provides students, staff, academics and developers with different software and hardware tools. However, in cloud computing all these arrangements can be migrated to the cloud (Sultan, 2010). Figure 2 illustrates an example of how cloud computing is used in the university.

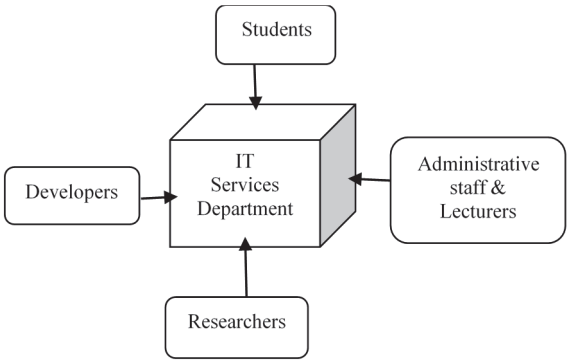

Figure 1: Users of Traditional IT services in a University (Sultan, 2010).

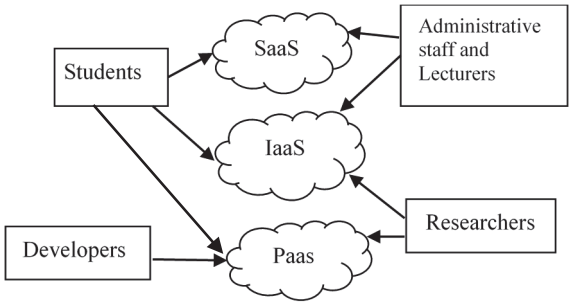

Figure 2: Cloud computing service models in a University (Mathew, 2012).

Cloud Computing offers services that enable the universities to concentrate more on teaching and research activities rather than building on complex IT configurations and software systems (Sultan, 2010). It can also be deployed more quickly. Complexity can be reduced with Cloud Computing. Students can exploit different learning tools. Students already use some, such as Google Docs and Office365 and Windows Azure Platform for computer science students (Ercan, 2010). Students can access the learning resources they need from anywhere and at any time with any Internet capable device.

Lecturers may experience flexible benefits as the cloud provides an easier platform to prepare their

teaching portfolio presentations, lessons, conferences, articles, etc. Researchers may also benefit from the advantages of using the latest technologies and hardware to do their experiments, while paying for using these services only on demand (Mircea and Andreescu, 2011).

Developers can design, build and test applications on the infrastructure of the cloud service provider and produce those applications from cloud provider data centres to the end user (Sultan, 2010; Huang, 2012) System administrators can leverage processing, storage, database management and other resources available on the cloud.

4 CLOUD SERVICES IN HIGHER EDUCATION

The trend of educational cloud computing has been adopted by many leading IT companies. Microsoft, Google, Amazon and IBM have provided much initiative to support education institutions with the necessary learning tools. Some of these initiatives are free with no cost. Table 1 shows some of the existing educational clouds and tools. With the availability of content online, it is unnecessarily for lecturers to print teaching materials. Nowadays, students have the choice to access homework assignments, lesson notes, and other materials online with the cloud. Some of the leading cloud services in higher education are described below.

- **Microsoft Education Cloud.** Microsoft Education Cloud has been actively developing educational cloud services such as Microsoft Office 365. It provides schools with free email, website with editing and storage facility, instant messaging, web conferencing, and 25 GB of personal storage (Jay, 2014). Furthermore, students and faculty are able to use any browser to create documents using Microsoft Office (David, 2013).

The downside to Microsoft 365 is the cost. While a free option is available (with a signed contract), a per-user monthly payment is required to access features such as Office Mobile, Office applications for PC or Mac, unlimited email storage and voicemail. More alarming is Microsoft's inability to ensure 99.9% uptime without monthly payment (Jay, 2014).

- **Google Education Cloud.** Google Apps for Education is one of the most used application as it does not involve actual cost (Jay, 2014). It is free with no hidden costs. Some of the feature include cloud email, 30GB of storage, hosting, word processing and collaboration tools (Google, 2015). Google is Microsoft's strongest competitor. If it is compared to Microsoft's Office Suite, there is an existing familiarity with many of Google's products such as Gmail, Chat, and Calendar. Nevertheless, the main drawback is that it requires users to have (or create) a Google account. It is compulsory for user of age 13 years old and below to get parent consent.

Table 1: Examples of educational cloud-based applications (Razak, 2009; Alshwaier, 2012).

Commercial Product Name	Education cloud apps	Features
Microsoft Education Cloud	Microsoft Live@edu	Website Creation File sharing Word processing Desktop sharing Resource scheduling
Google Education Cloud	Google Apps Education (GAE)	Google Mail Google Sites Google Docs Google Video Google Calendar Google Talk
Earth Browser	Earth Browser	Provide real Time data for weather, geological and other data
Socratica	Socratica	Classrooms in science to access Create and study modules
VMWare	Virtual Desktop	Provide Virtual computers
IBM Cloud Academy	Virtual computing lab	Smart analytics system

- **Earth Browser.** Earth Browser is a virtual globe software developed by Lunar software. It is available online as a flash application or be installed locally as an application (EarthBrowser, 2015a). It focuses mainly on visualising geophysical information such as weather, earthquakes, etc. It shows the earth as satellite images. EarthBrowser can be used in real-time. It is shows the object in three dimensional model with continuously updates information (EarthBrowser, 2015b). The representation of the earth is rendered along with a large data which is said to be accurate. The object can also be rotated and zoomed to a given distance.

- **Socratica.** Socratica produces high-quality educational videos for people of all ages (Socratica, 2015). The videos developed are high-definition, clear, concise, and beautiful. Socratica collects and organizes the best free educational videos into topics that can be used by users. Socratica's mission is to organise educational videos. This can be used by users to create optimised learning experience. They

have also restricted videos suitable for age groups by having different channels in YouTube.

- **Virtual Desktops.** In computing, a virtual desktop is known as another user interface that is able to provide user with the virtual space of a computer's desktop environment through the use of a software application installed in a user's physical computer. (VMware, 2015). Generally, there are two ways to expand the virtual area of the screen. The virtual desktop are switchable allowing user to create virtual copies of their desktop that is switchable. This can be done with open windows existing one desktops.

Another approach can expand the size of one virtual screen more than the physical viewing device. Usually, navigating an oversized virtual desktop is viewed using scrolling/panning into the subsection of the virtual desktop. One of the most popular VMware product is VMware Horizon 6. It provides a virtual desktop infrastructure (VDI) platform that provides virtualized and remote desktops and applications system through one platform, enabling users access to their online resources through one integrated workspace (VMware, 2015).

- **IBM Cloud Academy.** IBM cloud academy is a collaborative community of leaders in education. It is intended for educational institutions, with a goal to help reduce costs and optimise services while making information available, and secure if needed (IBM, 2014). It can also be used to consolidate resources, improve student success, and accelerate scientific discoveries. On the management part, it is expected to add administrative efficiencies, and conserve resources.

These are known as how cloud can help educational institutions to provide services. They are actively integrating cloud technologies into their infrastructures to share best practices in the use of clouds and to collaborate with partners to create innovative cloud technologies and models (IBM, 2014).

5 CLOUD COMPUTING ADOPTION CHALLENGES IN HIGHER EDUCATIONS

Despite the flexibility, scalability, on demand and powerful recourses cloud computing paradigm offers the higher education institutions, there is a low rate adoption of cloud computing in higher education institutions according to Gartner evaluation. Gartner mentioned that only 4% is the existing usage of cloud services in education. Another study highlights that 12% of the participants are not familiar with cloud computing services whereas 88% of them agree that cloud computing education services must be exploited in the schools (Kurelovi, Rako and Tomljanovi, 2013). However, migration to the cloud may not be an easy task overnight. The higher education institutions face several challenges that hinder adopting cloud computing. Researchers have highlighted many factors that affect universities' decisions to adopt cloud computing as shown in Table 2. The challenges are described in the section below.

5.1 Security

Security in cloud computing is a major concern faced in the adoption of cloud computing, not only in academic institutions but in all industries. Cloud providers must maintain confidentiality, integrity and availability (CIA) by establishing security requirements to satisfy educational cloud computing systems. Some of these requirements are identification and authentication accounts for students, faculty members and administration staff to verify and validate each individual by username and password. Some need control permissions, priorities and resource ownership (authorisation). Encryption techniques should be employed to protect sensitive data of institution such as exams, grades, etc. from tampering or unauthorized access. There is also need to ensure non-repudiation is some circumstances which means the transactions cannot be denied using time stamps, digital signatures and confirmation receipts (Razak, 2009; Ketel, 2014; Sultan, 2010; Mathew, 2012; Alshwaier, 2012).

5.2 Privacy

Privacy in higher education ensures sensitive data are protected from unauthorised and unauthenticated access in the cloud. Student's records, researchers' intellectual property should be maintained on the cloud. To protect the privacy of personal data European Union (EU), has privacy regulations that prohibit the transmission of some types of personal data outside the EU. This issue has required companies such as Amazon and others to provide offerings of storage facilities located in the EU. Regulation compliance impedes some higher education from adopting cloud-computing paradigm (Razak, 2009; Sultan, 2010; Mathew, 2012).

5.3 Lock-in

Vendor lock–in means that the university or instituti-

Table 2: Adoption Challenges in Higher Education.

Cloud Computing Adoption Challenges in Higher Education								
Authors	Security	Privacy	Lock-in	Reliability	Bandwidth	Management	Trust	Acceptance
(Abdul Razak, 2009)	√	√					√	
(Sultan, 2010)	√	√	√	√				
(Alshwaier, 2012)	√		√					
(Mathew, 2012)	√	√		√	√	√		
(Ketel, 2014)	√				√	√		√
(Shakeabubakar, 2015)				√			√	

on using cloud services from one provider may find all data they store and apps they use are locked-in to the products of specific provider which implies risks and significant costs to migrate to another vendor or to revert to on-premises traditional IT systems (Alshwaier, 2012; Sultan, 2010).

5.4 Reliability

Reliability has also been an issue for cloud users. For example, in February 2008, Salesforce.com customers were without service for 6 hours while Amazon's S3: simple storage service and EC2 experienced 3 hours outage in the same month a few days later and 8 hours outage in July. An outage is the absence of the Cloud service. Outage of the services in Higher education institutions can disrupt students from learning and can affect the learning schedule for the classes. It was mentioned that an 100% availability is impossible (Mathew, 2012; Sultan, 2010).

5.5 Bandwidth

Internet bandwidth is the backbone of the internet-based educational services. The quality of service relies on the connection speed, which can require investment in the network infrastructure (Ketel, 2014; Mathew, 2012).

5.6 Management

There are differences between traditional education management and education management with cloud computing. Hence, implementing cloud computing will lead to management challenges such as how to manage teaching and learning, the content and courses, the examinations and students (David, 2013; Ghorab, 1997).

5.7 Trust

Trust of online services is one of the most challenging factors in academia. In 2013, a research was conducted in Malaysian Universities (Shakeabubakor, 2015), which were UKM, UTM, UM, and UNITEN. The study aim was to investigate the researchers needs of productivity tools based on cloud computing in their reserch practices. The authors conducted interviews with researchers and postgraduate students. One of the significant findings was that 89% of the interviewed researchers distrust cloud application in their research activities (Shakeabubakor, 2015; Razak, 2009).

5.8 Acceptance

It is not easy to convince the decision makers in higher education to shift from one pattern to another. Cloud computing is a new IT Paradigm and it will change the familiar traditional pattern. Therefore, the users' (academics and top management) perception and acceptance will have an effect on adoption of cloud computing within institutions (Ketel, 2014).

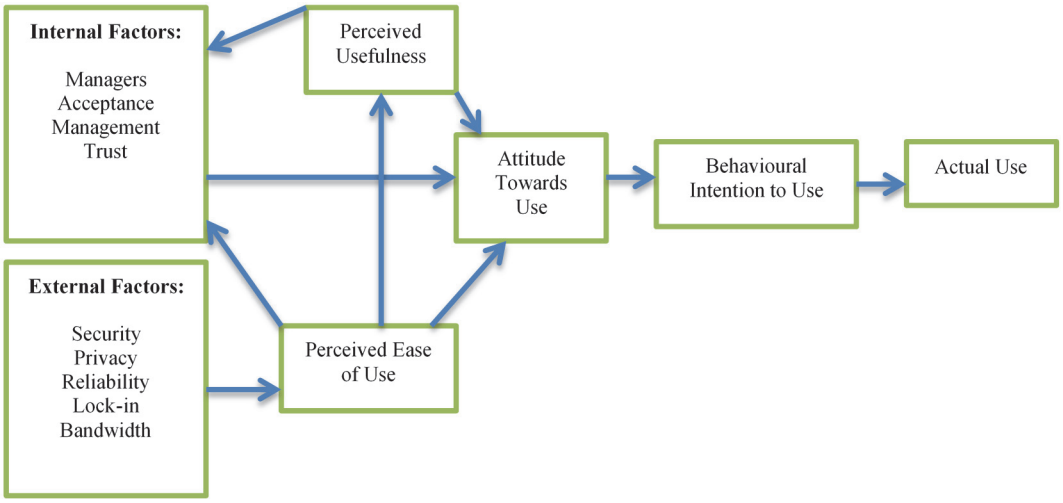

Figure 3: A proposed integrated model for cloud computing adoption in higher education institutions.

6 CLOUD COMPUTING ADOPTION MODEL IN HIGHER EDUCATION

In this paper, we suggest an investigation towards challenges aforementioned. Technology Acceptance Model (TAM) (Davis, 2014) is adopted and integrated with additional factors in the literature to investigate further factors that influence the adoption of cloud computing in higher education institutions. TAM is the most influential extensions of Ajzen and Fishbein's Theory of reasoned action (TRA) (Ajzen and Madden, 1986). Based on an examination of computer-usage behaviour, Davis developed the TAM, which is designed to predict acceptance of information technologies and use on the job. It has been widely applied to variety of technologies and users. Several researchers have replicated TAM model to provide empirical evidence on the existing correlation between the usefulness and ease of use when using new technology. In addition, the researchers focused on testing the validity and reliability of the questionnaire instrument used by Davis and they found that the instrument had predictive validity for intent to use, self-reported usage and attitude toward use with different samples of users and different technologies (Hendrickson et al., 1993; Albert, 1993; Szajna and Worth, 1994). The TAM addresses why users accept or reject the use of information technology due to external variables: in our model we categorise the external variable and divide them into two types:

1- **Internal Factors:** consist of Managers, decision makers and academic expert's acceptance and their cultural and social believes and training needs to use cloud services in higher education institutions 'user's factors'.
2- **External Factors:** consist of Security, privacy, reliability, lock-in and Bandwidth 'Technological factors' that intervene and indirectly affect their attitude toward using it.

In this theory, the individual's attitude is based on two elements; the first one**, 'perceived usefulness (PU)'** which is the measurement of the person's beliefs about whether using the cloud services in Higher education would enhance their job performance. Perceived usefulness is an important element for investigating individual acceptance of a new technology (Ghorab, 1997; Anandarajan et al., 2002). According to Davies (Davis, 2014), individuals tend to use an electronic system when they believe that using the system will help in improving their job. It was confirmed that perceived usefulness factor has a strong impact on e-learning success (Park, 2009). So, in this study the users in universities such as IT staff and academic are more likely to use cloud education services if they feel that it is useful in education purposes.

The second, 'perceived ease of use (PEOU)' is the measurement of the person's beliefs about using the cloud services in higher education institutions without expending extra effort. Perceived ease of use is defined as the extent to which the academic staff believe using cloud education emerging services would be free of effort. Perceived ease of use plays a key role in investigating individual acceptance of a new technology. TAM is the most widely applied

model of user acceptance and usage. When users feel the technologies can be used in an easy way, it is more probable that they will adopt cloud services in their educational practices, so ease of use will affect universities staff attitude and behaviour. Therefore, the factor is selected in the model to examine the users' acceptance of using cloud services in their teaching as academic or to store records and leverage different services such PaaS for developers to design, implement, test, and run new software.

If the ease of use and usefulness of cloud computing services in higher education has been recognised by academics and top management personnel. This may lead to an increase the adoption rate of cloud computing in the education sector. Figure 3 above shows the proposed integrated model for this context. This will be used as a reference model in investigating the adoption factors in higher education institutions.

7 CONCLUSION

Cloud computing in higher education is still in its infancy compared to other industries. However, over time it will continually grow. The adoption of cloud computing may help universities to focus more on their main goals which are related to teaching and learning with minimum expenditure. Students and staff can rapidly and cost-effectively access various application platforms and pool of resources on-demand. Cloud computing services are useful and sometimes necessary to meet challenges and barriers to providing IT services in Universities.

Important challenges include security, privacy and vendor lock-in that can affect the adoption of cloud computing in education but there internal factors such as user's acceptance, user's trust, Internet efficiency and the educational management roles. This is an ongoing research of challenges that affects the adoption of cloud computing in higher education. Based on previous research, there is a lack of empirical studies investigating the low adoption of cloud computing in higher education institutions. Our future work will focus on investigating success factors for adoption of cloud computing in higher education using the proposed integrated reference model in this paper.

REFERENCES

Abdul Razak, S. F., 2009. Cloud computing in malaysia universities. *2009 Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2009*, (July), pp.101–106.

Ajzen, I. and Madden, T. J., 1986. Prediction of goal-directed behavior: Attitudes, intentions, and perceived behavioral control. *Journal of experimental social psychology*, 22(5), pp.453–474.

Alabbadi, M., 2011. Cloud computing for education and learning: Education and learning as a service (ELaaS). *14th International Conference on Interactive Collaborative Learning (ICL2011)*, (September), pp.589–594.

Albert, H., 1993. Re-examining perceived ease of use and usefulness : A confirmatory factor analysi. 17(4), pp.517–525.

Alshwaier, A., 2012. A New Trend for E-Learning in KSA Using Educational Clouds. *Advanced Computing: An International Journal*, 3(1), pp.81–97.

Anandarajan, M., Igbaria, M. and Anakwe, U. P., 2002. IT acceptance in a less-developed country: A motivational factor perspective. *International Journal of Information Management*, 22, pp.47–65.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. and RH, 2009. Above the clouds: A Berkeley view of cloud computing. *University of California, Berkeley, Tech. Rep. UCB*, pp.07–013.

Buyya, R., Broberg, J. and Goscinski, A., n.d. *Cloud Computing Principles and Paradigms*.

David, W., 2013. *Google Vs. Microsoft: Choosing Cloud Apps For Schools - InformationWeek*.

Davis, F. D., 2014. Information Technology Introduction. 13(3), pp.319–340.

EarthBrowser, 2015a. *EarthBrowser*. [online] Available at: <http://blog.earthbrowser.com/> [Accessed 12 Mar. 2015].

EarthBrowser, 2015b. *EarthBrowser - Frequently Asked Questions*. [online] Available at: <http://www.earthbrowser.com/about/> [Accessed 12 Mar. 2015].

Ercan, T., 2010. Effective use of cloud computing in educational institutions. *Procedia - Social and Behavioral Sciences*, 2(2), pp.938–942.

Ghorab, K. E., 1997. The impact of technology acceptance considerations on system usage, and adopted level of technological sophistication: An empirical investigation. *International Journal of Information Management*, 17(4), pp.249–259.

Google, 2015. *Google Apps for Education*. [online] Available at: <https://www.google.com/work/apps/education/> [Accessed 12 Mar. 2015].

Hendrickson, A. R., Massey, P. D. and Cronan, T. P., 1993. On the Test-Retest Reliability of Perceived Usefulness and Perceived Ease of Use Scales. *MIS Quarterley*, 17(2), pp.227–230.

Huang, X., 2012. An E-learning System Architecture based on Cloud Computing. *Engineering and Technology*, pp.74–78.

IBM, 2014. *IBM Cloud Academy - Overview - United States*. Available at: <http://www.ibm.com/solutions/education/cloudacademy/us/en/> [Accessed 12 Mar. 2015].

Jay, G., 2014. *Edutech for Teachers » Blog Archive » Guest*

Post: Google vs. Microsoft: Cloud Apps for Educators.

Ketel, M., 2014. E-Ieaming in a Cloud Computing Environment. pp.0–1.

Kurelovi, E. K., Rako, S. and Tomljanovi, J., 2013. Cloud Computing in Education and Student ' s Needs. *MIPRO, Opatija, Croatia*, pp.726–731.

Mathew, S., 2012. Implementation of Cloud Computing in Education - A Revolution. *International Journal of Computer Theory and Engineering*, 4(3), pp.473–475.

Mell, P. and Grance, T., 2011. *The NIST Definition of Cloud Computing*. Gaithersburg, MD: National Institute of Standards and Technology (NIST).

Mircea, M. and Andreescu, A., 2011. Using Cloud Computing in Higher Education: A Strategy to Improve Agility in the Current Financial Crisis. *Communications of the IBIMA*, 2011, pp.1–15.

Motta, G., Sfondrini, N. and Sacco, D., 2012. Cloud computing: An architectural and technological overview. *Proceedings - 2012 International Joint Conference on Service Sciences, Service Innovation in Emerging Economy: Cross-Disciplinary and Cross-Cultural Perspective, IJCSS 2012*, pp.23–27.

Park, S. Y., 2009. An Analysis of the Technology Acceptance Model in Understanding University Students' Behavioral Intention to Use e-Learning. *Educational technology & society*, 12(3), pp.150–162.

Shakeabubakor, A., 2015. Cloud Computing Services and Applications to Improve Productivity of University Researchers. *International Journal of Information and Electronics Engineering*, 5(2), pp.153–157.

Socratica, 2015. *About Socratica*. [online] Available at: <http://www.socratica.com/about.html> [Accessed 12 Mar. 2015].

Sultan, N., 2010. Cloud computing for education: A new dawn? *International Journal of Information Management*, 30(2), pp.109–116.

Szajna, B. B. and Worth, F., 1994. Software Evaiuation and Choice : Predictive Vaiidation of the Technoiogy Acceptance instrument. 18(3), pp.319–325.

VMware, 2015. *Virtual Desktop Infrastructure (VDI) Features of Horizon (with View)*. [online] Available at: <http://www.vmware.com/uk/products/horizon-view/features.html> [Accessed 12 Mar. 2015].

AUTHOR INDEX

Proceedings of ESaaSA 2015

2nd International Workshop on Emerging Software as a Service and Analytics

www.closer.scitevents.org

ISBN: 978-989-758-110-6