Economics Models and Policies for Cloud Federations

George Darzanos, Iordanis Koutsopoulos, George D. Stamoulis
Athens University of Economics and Business, AUEB, Athens, Greece
{ntarzanos, jordan, gstamoul}@aueb.gr

***Abstract*—Cloud federation has emerged as an effective solution offering worldwide coverage, dynamic infrastructure scaling and improved QoS for the demanding cloud services. In this paper, we present a model for Cloud Service Providers (CSPs) federation and we investigate the economic benefits of CSPs under different federation modes. Each CSP is modeled as an M/M/1 queue with arrivals corresponding to computation tasks and service rate that captures the computational capabilities of the CSP. Each CSP earns revenue by charging its customers according to a QoS-dependent pricing function, and it undergoes a cost due to energy consumption of its infrastructure. We propose a model for the formation of cloud federations, according to which each CSP may forward part of the workload stemming from its customers to other CSPs. We define three federation modes with varying degrees of CSPs' interaction, namely the strong, weak and elastic federations. In strong federations, the CSPs jointly decide on their forwarding policies to maximize the total profit of the federation; then, they share these profits according to certain profit sharing policies. In weak federations, a game arises, in which each CSP follows a forwarding policy that aims to maximize its individual payoff, which however incorporates some fairness. In elastic federations, each CSP again aims to maximize its individual payoff, but it has the freedom to tune the degree of its selfishness through a pricing function. The numerical results validate and quantify the conjecture that federation can incur substantial monetary benefits and achieve a near to optimal QoS.**

I. Introduction

Nowadays, the multi-faceted applications that are moving to the cloud demands global geographic presence and high QoS for end-users. Although Cloud Service Providers (CSPs) promise flexible and scalable resources, thus creating the illusion of infinite resources to their customers, no CSP can provide on-demand dynamic resource scaling in order to handle the workload variations in a cost-effective manner. Furthermore, even market giants have limited geographic coverage since it is not profitable to invest on establishing datacenters in multiple geographical locations to satisfy the demand. Cloud federation arises as an effective way to expand the reach of CSPs and improve the QoS of their customers.

In a cloud federation, multiple CSPs cooperate to provide seamless provisioning of high-quality services across different domains. A cloud federation should be accompanied by certain policies that ensure the sustainability of this CSP community, and each CSP that participates has to conform to these policies. The policies should guarantee that each CSP that joins the federation will not undergo profit loss. Further, they need to motivate all CSPs to participate regardless of their market power or the size of their infrastructure. Cloud federation comes together with several participation incentives such as geographic footprint expansion, the scaling of resources to handle the request traffic bursts of peak demand and the inter-cloud load balancing. Hence, a cloud federation prevents the datacenter over-dimensioning and further it reduces the CSPs' energy cost through better utilization of their infrastructure.

In real life, there are several instances of cloud federation in both academic and enterprise environments. The OnApp Federation [1] is a network of CSPs running on the OnApp cloud management platform. The CSPs that join this federation may buy and sell capacity on demand through the OnApp market. Arjuna's Agility framework [2] is a dynamic federated cloud computing platform that is created from IT resources that are offered by autonomous, cooperating business parties within and beyond an enterprise, and under certain policies. EGI Federated Cloud [3] is a seamless grid of academic private clouds and virtualized resources, built around open standards and it focuses on the requirements of scientific community. BonFIRE [4] offers a federated testbed that supports large-scale testing of applications, services and systems over multiple, geographically distributed, heterogeneous cloud and network testbeds. Finally, the CERN Openlab project [5] aims to build a seamless federation among multiple private and public cloud platforms on OpenStack.

Several works in recent literature investigate the problem of resource allocation in cloud federations. These works can be classified into two broad categories: *(i) Cooperative resource pooling* [6]–[8], where CSPs aggregate their resources aiming to maximize the total utility of federation and *(ii) Resource trading* [9], [10], where CSPs aim to maximize their individual profit by trading their unused resources. In our prelude work [11], we modeled each CSP as an M/M/1 queueing system and devised a mathematical model for the net profit of each CSP. This consists of accrued revenues from pricing on its customers and of incurred energy cost at the cloud infrastructure. Furthermore, we introduced a first approach to model a service-oriented cloud federation between two CSP, where each CSP may forward a portion of the tasks stemming from its customers to other CSPs. Finally, we formulated the problem of finding the utility-optimal federation as a global profit maximization problem in which CSPs align their strategies to jointly solve it.

In this paper, we build on and substantially extend our previous work by studying different cloud federation regimes. In particular, we define the strong, weak and elastic federation

ISBN 978-3-901882-83-8 © 2016 IFIP

modes. Each mode differs on the level of cooperation among CSPs, the extent of private information that CSPs should make available to others, and the CSPs' objectives that may be aligned or conflicting. *Strong* federation requires an offline mutual commitment of CSPs, such that they all agree to align their forwarding policies to optimize their total net profit. Additionally, a mutually agreed policy is applied for the fair sharing of total profit of federation. In *weak* federation, the CSPs are still able to forward tasks, however each of them acts unilaterally by trying to maximize its own net profit, and thus an non-cooperative game arises. Nevertheless, the net profit of each CSP in this type of federation is strongly connected to its contribution in the federation, namely its profit share is given by its Shapley value. It will be seen that use of Shapley value as payoff function leads the relevant game to an efficient equilibrium. We also develop a more *elastic* model for cloud federations whereby all CSPs employ a flexible pricing scheme on forwarded tasks that reflects their degree of selfishness. Again, each of them aims to maximize its net profit and thus a non-cooperative game arises, the outcome of which depends on this degree of selfishness. For each federation mode we formulate the problem of net profit-optimal service delegation and we find the optimal forwarding policies.

The paper is organized as follows. In section II we provide an overview of relevant state-of-the art work. In section III, we present our model for a single CSP. In section IV we present our cloud federation model, we introduce and specify the three modes of federation and solve the relevant optimization problems. In section V we present our numerical evaluation, and in section VI we briefly present our conclusions.

II. Related Work

Architectural approaches of cloud federation. The authors in [12] present the challenges of a utility-oriented cloud federation and propose three basic entities for a market-based cloud federation architecture; the cloud exchange as the entity that creates the market, a cloud coordinator per CSP as seller and a cloud broker per client as buyer. The Reservoir model, a modular cloud architecture, is proposed in [13]. In Reservoir, multiple CSPs collaborate in order to create a virtual pool of resources that seems infinite. The authors in [14] present the concept of cloud federation as service aggregation and they present two modes of such a federation, the redundancy and migration federations. In redundancy federation, multiple CSPs come together and jointly offer a service to achieve improved quality for a client, while in migration federation a client is moved from an old service to new one offered by another CSP due to improved quality. Finally, the authors in [15] envision the federation of CSPs as vertical stack that fits on the layered model of cloud computing. A service request may arrive in any layer of a CSP and can served either by local resources using delegation to a lower layer or by another federated CSP using delegation to a matching layer.

Cooperative inter-cloud resource allocation. The authors in [16] propose cooperative price-based resource allocation mechanisms in dynamic cloud federation platforms, aiming to maximize the total utility of federation. In [6] and [7], coalitional game theory is applied as a mechanism for the dynamic formation of CSPs' federation. Both these papers have proposed algorithms that determine the optimal coalitions for a set of CSPs, given their client generated workloads. In [8] the inter-CSP VM migration is presented a solution to the problem of resource over-provisioning. The authors propose a global scheduler that decides whether a VM should migrate or shut down, thus aiming to CSPs utility maximization.

Resource allocation among selfish CSPs. In [9], the federation among geo-distributed CSPs is investigated. The authors design double-auction based algorithms for inter-cloud VM trading in federations of selfish CSPs. The authors in [17] model each CSP as a set of heterogeneous servers, each of them modeled as a queueing system. Then, they formulate the problem of resource allocation in a multi-CSP environment as a game among selfish CSPs, where each CSP aims to maximize its individual utility taking into account the customer SLAs. The author in [10] investigates the interactions among CSPs as a repeated game among selfish players that aim at maximizing their profit by selling their unused resources in a spot market. The model incorporates information for both historical and expected future revenue as part of the resource trading decision, in order to simultaneously maximize the CSP revenue and avoid future workload fluctuations.

Some of the above works provide an overview of the architectural elements of a federated system, while others consider the problem of resource allocation in inter-cloud environments of either cooperative or selfish CSPs. In our work, we propose the concept of federation among CSPs as service delegation and we model the federated environment as well as its involved economics. Contrary to most existing works, we provide policies both for cooperative and the non-cooperative federated environments. Further, we propose a flexible policy where CSPs can move between cooperation and selfishness. Additionally, most of existing works do not take into account the QoS offered to CSPs' customers in their optimization approach. In our work, the federation policies are optimal with respect to total or individual CSPs' profit (depends on policy), but they are also beneficial with respect to the QoS offered to customers.

III. Cloud Service Provider Model

A. CSP as an M/M/1 Queueing System

For each CSP i, we use C_i to denote the total computational capacity (in operations/sec) of its infrastructure. We assume that tasks from its customers arrive to a central controller according to a Poisson process of rate λ_i (tasks/sec). Each of these tasks requires a random number of operations in order to be executed. We assume that the number of operations follows an exponential distribution with mean number L operations/task. The average service rate (in tasks/sec) for a CSP i is $\mu_i = C_i/L$, and thus the service time of a task is exponentially distributed with mean $1/\mu_i$.

We use the *average task completion time* as a metric for customers' QoS. By standard theory for M/M/1 single-server queueing systems, the average completion time d_i for tasks served by the infrastructure of CSP i is given by

$$d_i(\lambda_i) = \frac{1}{\mu_i - \lambda_i}. \tag{1}$$

The average rate of incoming tasks must always be lower than the service rate of the system ($\lambda_i < \mu_i$), otherwise the CSP queue becomes unstable.

M/M/1 abstraction justification. In practice, a CSP consists of multiple datacenters with servers within each of them. In our approach, we abstract the multi-server infrastructure of the CSP as a single-server M/M/1 queueing system. To this end, we assume that a CSP performs perfect dispatching and scheduling of incoming tasks by preventing its servers from becoming idle. In particular, if the infrastructure of a CSP i consists of M_i identical servers of computational capacity C_i/M_i each, the CSP achieves the same average utilization level ρ_i to all servers by applying the optimal internal task dispatching and scheduling. Hence, we can safely assume that the multi-server infrastructure of each CSP behaves as a *single-server* with computational capacity C_i and utilization ρ_i.

The queueing-system assumption can be justified as follows: Tasks arrive at the controller in the form of a stream. Since cloud computing provides the technology for virtualizing resources, tasks coming from the customers of a CSP are assigned to established virtual machines (VMs). VMs are not in abundance, but they are finite resources that are assigned on-demand to serve requests. Given that a typical cloud computing system serves a large number of customers where each of them generates multiple computational tasks and these arrive in bursts, is more probable to have smaller interarrival times than larger ones. Thereafter, we can assume that the tasks arrive according to a Poisson process. Furthermore, the time that a task spends in the CSP's system depends both on the waiting and service time, i.e. on the number of existing tasks that wait to be served, on the availability of resources when the task arrives and on its size with respect to the number of operations it entails. The majority of tasks that arrive in a CSP queue usually demands a smaller number of operation, while relatively fewer tasks require a large number of operation. Hence, we can assume that the number of operations that a task requires is exponentially distributed, and therefore the service time also follows an exponential distribution. Consequently, the M/M/1 queueing model is applicable. While this is a simplification that allows the mathematical treatment of our paper, it is also reasonable enough to capture the reality.

B. CSP Economics

Energy consumption cost. We take the infrastructure energy consumption cost of a CSP as measure for its total cost. The power consumption of a server includes the power for its operation and the power that is required for supportive systems like cooling devices. However, according to prevalent state-of-the art literature [18], the total power consumption is a linearly increasing function of the utilization factor of the server, ρ. Specifically, the total power consumed is the sum of server's idle power and utilization factor-dependent dynamic power consumption. The former amount of power, W_0, is the power consumed when the server is powered on but does not serve any task. The latter one is linearly increasing in the server utilization ρ. If we denote by W_1 the power of a server when it is fully utilized (namely at $\rho = 1$), the range of power consumption is $[0, W_1 - W_0]$.

To estimate the total power consumption of a CSP, we take into account that its infrastructure consists of multiple servers. Since a CSP achieves the same average level of utilization ρ in all its servers (subsection III-A), idle and dynamic power consumptions of the entire infrastructure can be computed by aggregating the corresponding power consumption patterns of all servers. Consequently, if a CSP i has M_i servers, and if $W_{0,ij}$ and $W_{1,ij}$ denote the idle and total power consumption of the j-th server of CSP i, the aggregate power consumption of the CSP i in Watts is

$$\begin{aligned} W_i(\lambda_i) &= \sum_{j=1}^{M_i} W_{0,ij} + \frac{\lambda_i}{\mu_i} \sum_{j=1}^{M_i} \Big(W_{1,ij} - W_{0,ij} \Big) \\ &= W_{0,i} + \Big(W_{1,i} - W_{0,i} \Big) \frac{\lambda_i}{\mu_i}, \end{aligned} \tag{2}$$

where $W_{0,i}$ and $W_{1,i}$ denote the idle and total power consumptions of i's infrastructure. If i uses electricity at a price Z_i per KWatt·sec, the *cost of energy consumption* per unit of time is given by

$$E_i(\lambda_i) = W_i(\lambda_i)\ Z_i. \tag{3}$$

QoS-dependent Pricing. We assume that a CSP charges its customers based on the offered QoS-level and on load of received requests. Recall that we use average tasks completion time as measure for the QoS offered by a CSP. Thereafter, a CSP i sets a price per task according to a pricing function $p_i(\cdot)$, where $p_i(\cdot)$ is decreasing in average task completion time, d_i. This function should also be convex, because a marginal change in delay is perceived more by the customer for smaller values of the delay. Further, the average completion time of task is always lower-bounded by the expected service time $\beta_i = 1/\mu_i$. A function that satisfies the requirements above is

$$p_i(\lambda_i) = \frac{\beta_i}{d_i(\lambda_i)}\ Q_i, \tag{4}$$

where Q_i denotes the price per task that i charges for offering service in the best possible QoS, i.e. the expected service time β_i. In practice, the pricing function for each CSP is driven by the competition in the cloud market. In our approach, we assume that each CSP has made a decision offline on its pricing function that already takes into account this competition. Moreover, we assume that CSPs cannot adapt their pricing functions and also that their customers are committed by some contract and therefore they cannot change their serving CSP.

Revenue. The revenue of a CSP is generated from pricing on the tasks coming from its customers. Since CSP is also committed by some contract, we assume that the tasks arrive in its queue are always served. Consequently, the revenue rate in monetary units per unit of time for CSP i is given by

$$R_i(\lambda_i) = \lambda_i\ p_i(\lambda_i). \tag{5}$$

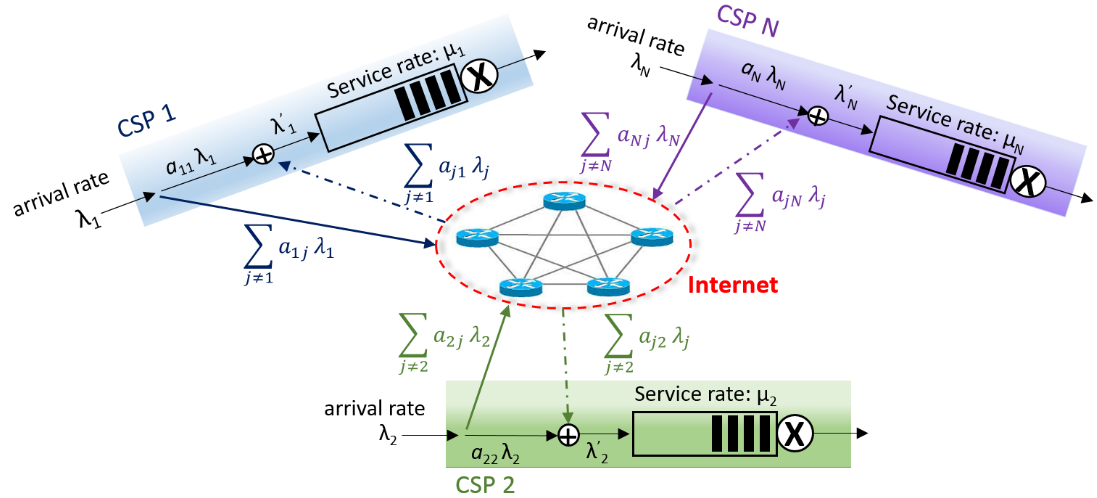

Fig. 1. Cloud federation for N CSPs, each of them is modeled as a single-server M/M/1 queue. Each CSP may forward a portion of the tasks coming from its customers to other CSPs and likewise it can receive task streams coming from customers of other CSPs. The streams of forwarded tasks between CSPs i and j undergo a fixed average transfer delay D_{ij}.

Net Profit. The net profit that i earns per time unit is

$$P_i(\lambda_i) = R_i(\lambda_i) - E_i(\lambda_i). \tag{6}$$

IV. Cloud Federation Policies

Our cloud federation model is based on the ability of each CSP to forward part of its incoming traffic stream of tasks to other CSPs within the federation. Therefore, the forwarding policy of each CSP is considered as its strategic leverage. The objective of a CSP for joining the federation may vary, and thus a CSP may have incentives to act either cooperatively or selfishly. We investigate three different modes under which the CSPs can federate:(i) the strong, (ii) weak and (iii) elastic federation modes. Each mode differs from others either in the level of private information that each CSP should make available to other CSPs or in the cooperation level of CSPs that may have common or conflicting federation objectives.

A. Model

We consider a set $\mathcal{N}$ of $N = |\mathcal{N}|$ CSPs, and for each CSP $i \in \mathcal{N}$ we define variables α_{ij} for $j = 1,..,N$ that determine the portion of its incoming tasks that CSP i forwards to a CSP j. Therefore, our global forwarding policy is a $N \times N$ dimensional matrix $\mathbf{A}$, whose entries α_{ij} determine the forwarding policy of all CSPs. We use vectors $\mathbf{a}_i$ and $\mathbf{a}'_i$ to denote the i-th row and i-th column of $\mathbf{A}$ respectively. The aggregate rate of tasks that CSP i forwards to others is $\sum_{j\in\mathcal{N}\setminus\{i\}} \alpha_{ij}\lambda_i$, while the average rate of tasks that CSP i receives from other CSPs is $\sum_{j\in\mathcal{N}\setminus\{i\}} \alpha_{ji}\lambda_j$.

Fig. 1 depicts our federation model for N CSPs. We assume that the portion of tasks that are transferred from a CSP to another, experiences an additional delay due to the intervening Internet links between their datacenters. Therefore, for each pair of CSPs $i, j \in \mathcal{N}$ we define an average communication delay D_{ij}. This delay is understandably exogenous to the system of CSPs. Also, we assume that the tasks that arrive in all CSPs belongs to the same service class and thus have the same mean number of operations, L, per task.

In our model, the task arrival rate at the input of each CSP's queue depends on the forwarding policy of other federated CSPs. Therefore, the ultimate arrival rate of tasks in the queue of CSP i depends on values of i-th column of matrix $\mathbf{A}$ (i.e. on vector $\mathbf{a}'_i$) and is defined as $\lambda'_i(\mathbf{a}'_i) = \sum_{j\in\mathcal{N}} \alpha_{ji}\lambda_j$. Thus, the average completion time of the tasks that are served by the infrastructure of CSP i is

$$d_i(\mathbf{a}'_i) = \frac{1}{\mu_i - \lambda'_i(\mathbf{a}'_i)}. \tag{7}$$

A portion of the task stream that arrives in a CSP is served by its own infrastructure, while other portions may be forwarded to other CSPs. Hence, the average completion time of tasks coming from the customers of CSP i depends on the average delay experienced at other CSP queues. Thus the average task completion time for customers of CSP i depends on all columns of matrix $\mathbf{A}$ and is defined as:

$$T_i(\mathbf{A}) = \sum_{j\in\mathcal{N}} \alpha_{ij}\big(d_j(\mathbf{a}'_i) + D_{ij}\big). \tag{8}$$

Note that $D_{ii} = 0$. At this point, it is important to stress the difference between $T_i(\cdot)$ and $d_i(\cdot)$:

- $d_i(\cdot)$ the average completion time for tasks that are served by i's infrastructure, including tasks originating from customers of i and tasks from other CSPs' customers.
- $T_i(\cdot)$ the average completion time of tasks that are generated from customers of CSP i, regardless of whether they are ultimately served by CSP i or by other CSPs.

In Section III a complete characterization of a single CSP is provided, however we need to slightly revise our model in order for it to be applicable in the federation. Now, the power consumption of i's infrastructure is affected by forwarding policies of CSPs. Thus, the power consumption of i's infrastructure is given by

$$W_i(\mathbf{a}'_i) = W_{0,i} + \big(W_{1,i} - W_{0,i}\big)\frac{\lambda'_i(\mathbf{a}'_i)}{\mu_i}. \tag{9}$$

Accordingly, the energy cost per unit of time is defined as

$$E_i(\mathbf{a}'_i) = W_i(\mathbf{a}'_i)\ Z_i. \tag{10}$$

The customers of CSP i should be charged based on $T_i(\cdot)$ rather than $d_i(\cdot)$ because different tasks may be served from different CSP queues. Hence, the pricing function becomes $p_i(\mathbf{A}) = \frac{\beta_i}{T_i(\mathbf{A})}Q_i$, and thus the revenue per unit of time is

$$R_i(\mathbf{A}) = \lambda_i \frac{\beta_i}{T_i(\mathbf{A})}Q_i. \tag{11}$$

Finally, the generated profit per unit of time is given by

$$P_i(\mathbf{A}) = R_i(\mathbf{A}) - E_i(\mathbf{a}'_i). \tag{12}$$

B. Strong Federation

All CSPs that participate in a strong federation comply to certain cooperation rules that have been agreed a priori. These rules include: (i) cooperation on exchanging private information, i.e. the values of their computational capacity C_i and average request load λ_i, (ii) agreement on the common objective of total federation profit maximization (iii) cooperation on defining the appropriate policy for sharing the total profit incurred from federation, and (iv) commitment to always serve the forwarded tasks of other federated CSPs.

Total Profit maximization. The CSPs cooperate and jointly decide the forwarding policies $\mathbf{A}$ that maximize the total federation profit. The globally optimal forwarding policy $\mathbf{A}^*$ is derived by solving the total profit maximization problem,

$$\begin{array}{lll} \arg\max\limits_{\mathbf{A}} & \sum\limits_{i\in\mathcal{N}} P_i(\mathbf{A}) & \\ \text{s.t.} & \alpha_{ij} \geq 0 & , \quad \forall i,j \in \mathcal{N}, \\ & \sum\limits_{j\in\mathcal{N}} \alpha_{ij} = 1 & , \quad \forall i \in \mathcal{N}, \\ & \lambda'_i(\mathbf{a}'_i) < \mu_i & , \quad \forall i \in \mathcal{N}. \end{array} \tag{13}$$

The second constraint captures the splitting of CSP i's task traffic across others. The third constraint is due to stability in the queues of each CSP. We can solve this non-linear problem by applying standard optimization methods, i.e. formation of the Lagrangian and statement of the necessary and sufficient KKT conditions that should be satisfied for optimality.

Profit Sharing Policies. Our problem formulation guarantees that under the optimal $\mathbf{A}^*$, the total federation profit is maximized. Thereafter, in the worst case scenario, i.e. in $\mathbf{A}^* = \mathbf{I}$ (Identity matrix), the total profit of federation equals the aggregate profit of CSPs in standalone operation. By standalone, we mean that each CSP serves only the tasks coming from its customers. However the individual profit may in fact deteriorate for one (or more) CSPs due to task forwarding actions. Specifically, the CSPs that only receive forwarded tasks may have loss because the extra workload will increase their energy cost due to the higher infrastructure utilization. As a result, these CSPs may be unwilling to comply with the federation, unless some rule is applied for the elimination of their losses. Since the total profit of the federation exceeds the aggregate profit of CSPs in the standalone mode, CSPs that only forward tasks definitely have higher profit than before, thus they are able to compensate others. Therefore, the CSPs have to reach an agreement for the fair sharing of total generated profit that satisfies all of them.

Next, we present two cooperative profit-sharing policies that serve the objective above. In the first policy, the profit share that a CSP receives depends both on its standalone profit and on the percentage of total forwarded tasks that it forwards or receives. In the second policy, we determine the profit that a CSP should get based on its marginal contribution in the federation by making use of Shapley value notion [19].

1) Interaction driven profit-sharing: In this approach, a CSP i gets at least the profit it had in standalone operation, while the extra profit generated from the federated operation is proportionally shared among N CSPs based on the percentage of forwarded tasks that each of them forwarded or received. We define the *extra generated profit* $P_F(\mathbf{A}^*)$ by subtracting the aggregate profit of CSPs in the standalone operation from the total profit of federation

$$P_F(\mathbf{A}^*) = \sum_{i\in\mathcal{N}} P_i(\mathbf{A}^*) - \sum_{i\in\mathcal{N}} P_i(\mathbf{I}). \tag{14}$$

where $P_i(\mathbf{I})$ denotes the profit of CSP i in standalone operation. Consequently, the share of CPS i is determined by:

$$\xi_i(\mathbf{A}^*) = \frac{|\lambda'_i(\mathbf{a}'^*_i) - \lambda_i|}{\sum\limits_{j\in\mathcal{N}} |\lambda'_j(\mathbf{a}'^*_j) - \lambda_j|} P_F(\mathbf{A}^*) + P_i(\mathbf{I}), \tag{15}$$

where $\frac{|\lambda'_i(\mathbf{a}'^*_i) - \lambda_i|}{\sum_{j\in\mathcal{N}} |\lambda'_j(\mathbf{a}'^*_j) - \lambda_j|}$ is the *proportionality* parameter which defines that a CSP who forwards or receives more tasks compared to another, will receive proportionally larger share of the extra generated profit.

2) Shapley value driven profit-sharing: Shapley value has been widely used in coalitional game theory applications as a mechanism for sharing total utility in a fair manner. A characteristic function $v(\cdot)$ measures the benefit of a coalition, also called the *worth of coalition*. In our approach, we take as characteristic function the total profit that is generated from the federated operation of a given set of CSPs. For instance, the worth of coalition $v(\cdot)$ for the set of $\mathcal{N}$ CSPs is

$$v(\mathcal{N}, \mathbf{A}) = \max_{\mathbf{A}} \sum_{i\in\mathcal{N}} P_i(\mathbf{A}), \tag{16}$$

where the solution is obtained by (13). For a federation of N CSPs, the Shapley value of each CSP is obtained by calculating its average marginal contribution in all possible sub-federations $\mathcal{S} \subseteq \mathcal{N}$. Therefore, we need to know the *worth of coalition* $v(\mathcal{S}, \mathbf{A}_S)$ for all possible subsets of CSPs $\mathcal{S}$. Note that $S = |\mathcal{S}|$ and $\mathbf{A}_S$ is the corresponding $S \times S$ dimensional matrix of forwarding policies. In order to find the worth of subset $\mathcal{S}$, we have to solve the relevant optimization problem (13) for all possible such subsets.

Assuming that $\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}$, the *marginal contribution* of CSP i when it joins a sub-federation $\mathcal{S}$ is defined as

$$\mathcal{MC}_i(\mathcal{S}, \mathbf{A}_S, v) = v(\mathcal{S} \cup i, \mathbf{A}_{\mathcal{S}\cup i}) - v(\mathcal{S}, \mathbf{A}_{\mathcal{S}}) \qquad (17)$$

Consequently, the profit share of a CSP i in the federation of N CSPs is given by its *Shapley value* defined as

$$\varphi_i(\mathcal{N}, \mathbf{A}) = \sum_{\mathcal{S} \subseteq \mathcal{N}\setminus\{i\}} \frac{|\mathcal{S}|!\big(N - |\mathcal{S}| - 1\big)!}{N!} \mathcal{MC}_i(\mathcal{S}, \mathbf{A}_S, v) \ , \quad (18)$$

where $\varphi_i(\mathcal{N}, \mathbf{A})$ denotes the *estimated marginal contribution* of CSP i over all possible subsets of $\mathcal{S}$.

Remark I. The two profit-sharing policies differs on how they perceived the level of a CSP's contribution. In interaction driven policy the extra profit is distributed only among the CSPs that are involved in forwarding actions of optimal policy, either as source or destination. On the other hand, Shapley value is less tight since it takes also into account the potential contribution of a CSP in all possible sub-federations. For more than two CSPs the policies may lead to totally different result.

C. Weak Federation

Weak and strong federation both require a level of commitment for each CSP in serving the requests forwarded to it by others. However, in weak federation the CSPs do not share the same objective any more, i.e. the maximization of total profit. Each of them determines its individual forwarding policy aiming to maximize its net profit, and thus a *non-cooperative game arises*. Since the CSPs have conflicting objectives, it is not sufficient to define the individual profit of each CSP as its payoff function as if the CSP were standalone. Otherwise, a selfish CSP would be able to outsource tasks without cost, taking the game to an equilibrium point where one or more CSPs may have less profit compared to that in their standalone operation. As a result, CSPs that undergo losses may be unmotivated to participate. In order to tackle this *participation constraint* and to simultaneously achieve a fair allocation of profits, it is announced to CSPs that their profit in federation is determined by a fair contribution-based profit sharing rule, namely their Shapley value. Then the CSPs are left alone to choose their own forwarding policies.

Non-cooperative Game. The set of players in this game is $\mathcal{N} = (1, 2, ..N\big)$. The individual forwarding strategy of a CSP i is defined by the entries of i-th row of forwarding matrix $\mathbf{A}$, thus the set of strategies of CSPs is $\mathcal{A} = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N)$. Note that $\mathcal{A}$ contains the same elements as $\mathbf{A}$. We define by $\mathbf{a}_i$ the strategy of CSP i, and by $\mathbf{a}_{-i}$ the strategies of all other CSPs except i. The payoff of its CSP in the game is determined by its Shapley value, thus the set of payoffs under a set of given strategies $\mathcal{A}$ is $\varphi = (\varphi_1(\mathcal{N}, \mathcal{A}), \varphi_2(\mathcal{N}, \mathcal{A}), ..., \varphi_N(\mathcal{N}, \mathcal{A}))$.

The game starts with each CSP operating in the standalone mode, where $\mathbf{A} = \mathbf{I}$. In every step of the game, a CSP i takes as input the current forwarding policies of other CSPs $\mathbf{a}_{-i}$ and determines its best response. The best response of CSP i is to select a forwarding policy $\mathbf{a}_i$ that maximizes its payoff $\varphi_i(\mathcal{N}, \mathbf{a}_i, \mathbf{a}_{-i})$. Therefore, i determines its *best response* by solving the following optimization problem:

$$\begin{aligned} \arg\max_{\mathbf{a}_i} \quad & \varphi_i(\mathcal{N}, \mathbf{a}_i, \mathbf{a}_{-i}) \\ \text{s.t.} \quad & \alpha_{ij} \geq 0 \quad , \quad \forall j \in \mathcal{N}, \\ & \sum_{j\in\mathcal{N}} \alpha_{ij} = 1, \\ & \lambda_i^{'}(\mathbf{a_i'}) < \mu_i. \end{aligned} \qquad (19)$$

In order to calculate its Shapley value, a CSP has to compute its marginal contribution in all possible sub-federations $\mathcal{S} \subseteq \mathcal{N}$. For the moment, we assume that this information is available and the game is played only for the full set $\mathcal{N}$ and not for subsets $\mathcal{S}$. At the end of this paragraph we elaborate on how the marginal contribution of CSP i in each $\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}$ can be obtained. The game continues until the system reaches a *Nash equilibrium* (NE) $\mathcal{A}^*$, where $\forall i \in \mathcal{N}$ and for every possible strategy $\mathbf{a}_i$, $\varphi_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*) \geq \varphi_i(\mathcal{N}, \mathbf{a}_i, \mathbf{a}_{-i}^*)$.

Claim. *Given a set of forwarding strategies $\mathcal{A}$. If CSP i applies a forwarding strategy $\mathbf{a}_i^*$ that maximizes its payoff under Shapley value objective function, $\mathbf{a}_i^*$ is globally optimal.*

Proof: Given that under strategy $\mathbf{a}_i^*$ the $\varphi_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*)$ of CPS i is maximized. Due to strong monotonicity of Shapley value [19] $\mathcal{MC}_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*, v)$ is also maximized. From (17), $\mathcal{MC}_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*, v)$ is maximized when the total profit of subset that i joins is maximized. Consequently, all CSPs adapt their forwarding policies in a such way that the total profit of federation is maximized.■

Corollary. *Under Shapley value payoffs the set of individually optimal forwarding strategies $\mathcal{A}^*$ is a Nash Equilibrium.*

Proof: We assume that in a step of the game all CSPs have chosen their optimal forwarding strategies $\mathcal{A}^*$ that according to our Claim are also globally optimal. We change the strategy of CSP i from $\mathbf{a}_i^*$ to $\mathbf{a}_i$, and we let the game continue. In the next step, CSP i changes back its strategy to $\mathbf{a}_i^*$ in order to maximize its payoff. Suppose that $\mathcal{A}^*$ is not a NE, there exist a CSP i that by changing its strategy to $\mathbf{a}_i$ can achieve $\varphi_i(\mathcal{N}, \mathbf{a}_i, \mathbf{a}_{-i}^*) > \varphi_i(\mathcal{N}, \mathbf{a}_i^*, \mathbf{a}_{-i}^*)$. However, since $\mathbf{a}_i^* \in \mathcal{A}^*$ this is a contradiction.■

Remark II. In order to determine its best response in previously presented game, each CSP should calculate its Shapley value based on its marginal contribution in all sub-federations $S \subseteq \mathcal{N}$. There are two alternatives to obtain this information: (i) The CSP plays recursive non-cooperative games as the above one for all the possible sub-federations. It starts playing these games from the smallest to largest subset, and the output of each game is used as input to the larger ones. (ii) Same as in section IV-B2, the CSP solves the relevant global optimization problem (13) for all subsets $\mathcal{S}$ and uses the results as input on determination of its best response in (19). Note that the second approach is less complex because we only have one game, however it has the *drawback* that CSPs should reveal their *private information* as done in strong federation.

D. Elastic Federation

The weak federation does not give CSPs the freedom to select their objective function. In this section, we propose the elastic federation where CSPs are free to tune their level of selfishness in the federation, and thus make a choice on their objective function. The elastic federation does not require any cooperation of CSPs on exchanging private information or on deciding a fair profit sharing policy. Each CSPs advertises a price that will charge all other CSPs for serving each forwarded task. Given these, each CSP decides its individual forwarding policy aiming to maximize its profit. Again, the CSPs have conflicting objectives and thus a non-cooperative game arises. However, by setting an arbitrary price per task, there is no guarantee that in the Nash equilibrium point the individual profit of each CSP will be better than its profit in standalone operation. Thus, we provide a rule for computing prices that does attain this goal.

Inter-CSP pricing rule. This rule guarantees that each CSP i sets a price that does not violate the federation participation constraint, i.e. CSP's i profit does not decrease due to federation. This is achieved by setting a lower bound on the price of each asking. This bound is determined based on an estimate of the negative impact that a forwarded task can have on the destination CSP's profit. In addition, the pricing rule gives CSPs the freedom to be as aggressive as they wish on the selection of price. In particular, CSP i sets the price per task by following the two steps below:

1) Lower bound of price: Given that the tasks arrival rate of CSP i in standalone operation is λ_i, we estimate the profit loss that a CSP would have by accepting to serve free of charge a number of $\mu_i - \lambda_i$ more tasks so as to reach utilization factor equal to 1. The profit is affected both by the increased energy consumption cost and by QoS degradation that brings revenue losses due to price reduction. When the utilization factor reaches 1, the average completion time $d_i \rightarrow \infty$, thus the price per task (4) becomes zero and the revenue is zero. Therefore, the revenue loss that a CSP can have equals to its revenue in standalone operation $R_i(\lambda_i)$. On the other hand, the additional energy cost for serving the number of additional tasks $\mu_i - \lambda_i$ is given by subtracting its energy cost in standalone operation from the energy cost that it would have in utilization level 1. Thus, the energy loss is given by $E_i(\mu_i) - E_i(\lambda_i)$. Consequently, the profit loss of CSP i for accepting $\mu_i - \lambda_i$ more tasks without charging is given by $R_i(\lambda_i) + E_i(\mu_i) - E_i(\lambda_i)$. Consequently, we can estimate the empirical per-task average negative impact by dividing the profit loss among the number of possible additional tasks, $\mu_i - \lambda_i$. Then, CSP i can set a lower bound on price $x_i(\lambda_i)$ per task that covers its profit loss, where

$$x_i(\lambda_i) = \frac{1}{\mu_i - \lambda_i}\Big(R_i(\lambda_i) + E_i(\mu_i) - E_i(\lambda_i)\Big). \tag{20}$$

2) Selfishness aware pricing: Having set the lower bound in the price, we now introduce the selfishness of each CSP in price setting, i.e. the level of its intrinsic desire to generate more revenue. The selfishness level of each CSP i is determined by a selfishness factor $\theta_i \in [0, 1]$, where $\theta_i = 0$ means that CSP i is not selfish and acts as being federation-friendly, and $\theta_i = 1$ implies that CSP i is extremely selfish. An extremely selfish CSP i would charge each task with a price that corresponds to the price paid by its customers in standalone operation, i.e. the price given from (4) for the current λ_i. On the contrary, a federation-friendly CSP would only charge a price $x_i(\lambda_i)$ per task.

In practice the parameter θ_i determines how the extra generated profit from a forwarded task is shared among the source and destination CSPs. If the destination CSP i is totally selfish, it gets all the extra generated profit; on the other hand if i is totally friendly, all the generated profit is gathered from the source CSP. However, being extremely selfish may discourage others from forwarding tasks toward i and select other more friendly destinations. Therefore, higher selfishness does not necessarily mean higher revenue. Based on the above analysis, the final price per task is determined as

$$\omega_i(\lambda_i) = x_i(\lambda_i) + \theta_i p_i(\lambda_i), \tag{21}$$

where $p_i(\lambda_i)$ is the QoS-dependent pricing function (4). In this paper, we assume that θ is fixed and same for all CSPs. The selection of optimal θ_i per CSP gives rise to new game-theoretic aspects that we plan to study in the near future.

Non-cooperative Game. In elastic federation the *payoff* of a CSP i includes its individual profit, the monetary amount that i receives by charging others for serving their tasks, and the monetary amount that i pays to others for serving tasks of its customers. Hence, the payoff of CSP i is defined as

$$\psi_i(\mathbf{A}) = P_i(\mathbf{A}) + \sum_{j \in \mathcal{N} \setminus i} \alpha_{ji}\lambda_j\omega_i(\lambda_i) - \alpha_{ij}\lambda_i\omega_j(\lambda_j). \tag{22}$$

Same as in weak federation, the set of *players* is $\mathcal{N}$ and their *strategies* are $\mathcal{A} = (\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_N)$, but now their payoff set is $\psi = (\psi_1(\mathcal{A}), \psi_2(\mathcal{A}), ..., \psi_N(\mathcal{A}))$. The *best response* of CSP i is given by the solution of $\arg\max_{\mathbf{a}_i} \psi_i(\mathbf{a}_i, \mathbf{a}_{-i})$ and under the same constraints as (19). The game stops when the CSPs converge to a Nash equilibrium point. The pure NE existence is confirmed by Debreu-Glicksberg-Fan's theorem [20].

V. Numerical Evaluation

A. Simulation setup

We focus our attention to the scenario of two CSPs in order to better understand and interpret the obtained results. We assume that the tasks that arrive in both CSP queues require an average of $L = 200$ Giga operations in order to be executed. We assume that both CSPs are symmetric with respect to computational capacity of their infrastructures $C = 2$ Tera operations per second. This capacity corresponds about 100 servers. The additional communication delay D for the forwarded tasks is taken to be an order of magnitude lower than tasks completion time in each CSP queue, $D = 0.01$. For the power consumption, we take the idle and total powers as $W_0 = 60$ KWatt and $W_1 = 400$ KWatt. Both CSPs pay the same price to their electricity provider, namely $Z = 2.7 \cdot 10^{-5}$ \$/KWatt·sec. Further, they both charge their customers according to the same pricing function, with same maximum price Q \$/task. In our experiments, we select the value of Q by taking as input the electricity price Z. In particular, given the price

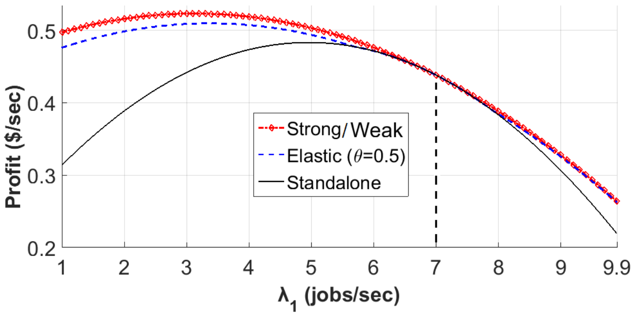

Fig. 2. Total profit of CSPs under different operation modes, for $\lambda_2 = 7$ and $\lambda_1 \in [1, 9]$

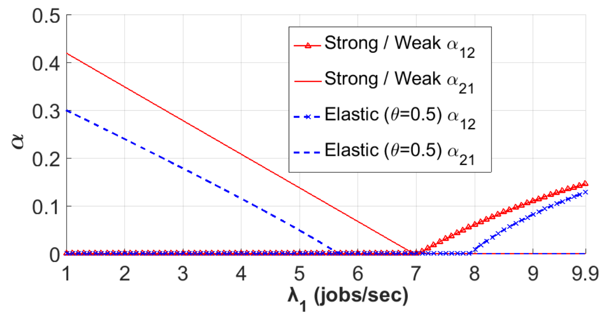

Fig. 3. Optimal forwarding policies of CSPs under different operation modes, for $\lambda_2 = 7$ and $\lambda_1 \in [1, 9]$

Z, we find the value of Q for which the profit of CSP becomes zero when the utilization factor is 0.99. This guarantees that both CSPs in standalone operation will not have negative profit for any value of utilization up to 99%. The price per task in our setup is $Q = 0.11$ \$/task. For the selfishness factor of CSPs in the case of elastic federation, we try out different combinations of θ values in the interval $[0, 1]$. In all experiments, we assume that CSP 2 has a fixed rate of incoming tasks λ_2 and we set values for λ_1 in the feasible range of values $[1, 9.9]$, with a step of 0.1. We run this type of experiment for different fixed values of λ_2 from 1 to 9.9.

B. Numerical Results

Total Profit. Fig. 2 shows the total profit under all operation modes, for fixed value of $\lambda_2 = 7$ and $\lambda_1 \in [1, 9.9]$. The results reveal that all three federation modes can achieve higher or at least the same total profit compared to the aggregate profit of CSPs in standalone operation. The total profit of strong and weak federation appears to coincide in all possible values of λ_1 and λ_2. This happens because of Shapley value's selection as a CSP's payoff in weak federation, since Shapley value urges each CSP to act for the benefit of all federation. In Fig. 2 we can observe that for $\lambda_2 = 7$ and for low load λ_1, strong and weak federation achieve a profit that is around $80-200\%$ more than the aggregate profit of CSPs in the standalone operation. For medium and high load, the benefit of strong and weak federation seems to diminish, while for $\lambda_1 = \lambda_2$, the total profit is equal to the one of standalone operation. Note that if the value of λ_2 were fixed to 9.9, the benefit of federation would be even higher for low and medium values of λ_1; about $100-400\%$ more than standalone. Consequently, the more diverse the CSPs' infrastructure utilization, the more pronounced the benefit of strong and weak federation is.

The total profit of elastic federation is strongly dependent on the selfishness factors θ of the federated CSPs. The results in Fig. 2 show that the total profit of elastic federation for $\theta = 0.5$ is lower but close enough to the one of strong and weak federation. Further, the results reveal that when θ is very close to or equal to 1, the extremely selfish CSPs set high prices and therefore the benefit of federation is eliminated. On the other hand, when θ equals to zero the total profit of elastic federation coincides with the total profit of strong and weak federation. Finally, the results show that for same value of θ, the benefit of elastic federation is relatively closer to strong and weak in high level of utilization, e.g. for $\lambda_1 = 9.9$ and $\lambda_1 = 7$ in Fig. 2.

Forwarding strategy. Fig. 3 shows the optimal forwarding policy of both CSPs under different federation modes. Interestingly, in the optimal solution at least one of α_{12} and α_{21} equals to zero. Further, the non-zero value always refers to the most utilized CSP. Strong and weak federation result to the same optimal pair $(\alpha_{12}^*, \alpha_{21}^*)$. In elastic federation, the value of non-zero α parameter is affected by the selfishness factor θ of the less loaded CSP which eventually receives the forwarded tasks. If $\theta = 0$, then the optimal pair $(\alpha_{12}^*, \alpha_{21}^*)$ of elastic federation is the same as in strong and weak federation. On the other hand, if the $\theta = 1$, the source CSP has no benefit from forwarding any task. Thereafter, the optimal pair of strong and weak federation is the upper bound for the optimal forwarding strategy of elastic federation. Further, we conducted additional numerical evaluations by setting different values in the communication delay D. The numerical results reveal that as the communication delay increases, the CSPs follow a more conservative forwarding policy and when D exceeds a certain value, the optimal pair becomes $(\alpha_{12}^*, \alpha_{21}^*) = (0, 0)$. Consequently, network delay is an important parameter for the effectiveness of federation.

Individual Profit. Fig. 4 and Fig. 5 show the individual profit of both CSPs under all possible operation modes. The individual profit of each CSP in all federation modes is higher or at least equal to its profit in standalone operation. The individual profit of a CSP under the interaction-driven profit sharing policy of strong federation equals its profit share when the Shapley value-driven policy is applied. However, this would be different in an experiment with more than two CSPs. In weak federation the individual profit of each CSP equals to its profit share in the strong federation. This happens because of Shapley value selection as payoff function of each CSP in the game. The individual profit of CSPs in elastic federation varies and is again related to their selfishness factor. In particular, for $\theta = 0$ a CSP that forwards a number of tasks gain all the extra revenue generated from that action, while the destination CSP only cover its profit loss. As θ increases the destination CSP demands a share of this extra generated revenue, therefore the profit share of destination CSP increases, and that of source CSP decreases. There are

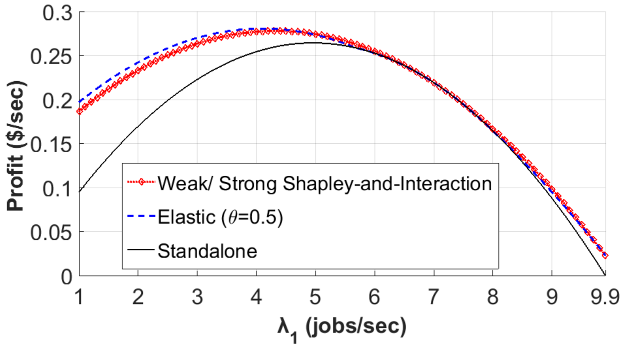

Fig. 4. Individual profit of CSP 1 under different operation modes, for $\lambda_2 = 7$ and $\lambda_1 \in [1, 9]$

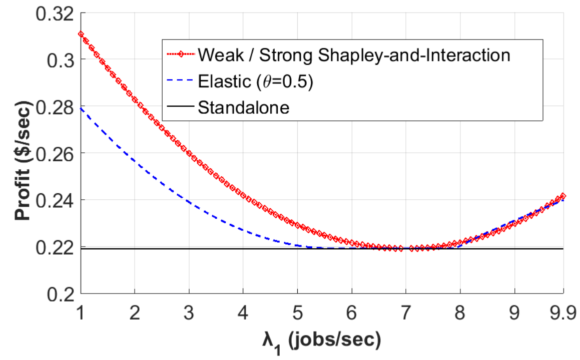

Fig. 5. Individual profit of CSP 2 under different operation modes, for $\lambda_2 = 7$ and $\lambda_1 \in [1, 9]$

non-zero values of θ where either CSP 1 or CSP 2 earns higher individual profit than in strong federation, however this cannot hold for both CSPs simultaneously because their aggregate profit cannot exceed the total profit of strong federation. A value of θ that achieves individual profit for both CSPs that are close to their profit in strong federation varies and depends on the input loads of CSPs. Consequently, the value of θ is debatable and needs further investigation.

QoS level. The results show that all federation modes outperform standalone operation and achieve a close-to-optimal QoS. The strong and weak federation achieves the same average task completion time, and further their performance is extremely close to the average completion time of a QoS-optimal federation. In elastic federation, for $\theta = 0$ the average task completion time equals the one of strong and weak federation, while for $\theta = 1$ elastic federation has the same performance as standalone operation.

VI. Conclusions

In this paper, we have presented models and policies for the formation of service-oriented cloud federations. Our models guarantee the economic sustainability of cloud federations both for cooperative and non-cooperative environments. The results show that in strong and weak federations the net profit of federation is maximized, while the offered QoS is very close the optimal one. The elastic federation gives CSPs the freedom to select their selfishness level. By selecting the appropriate selfishness level, a CSP may earn a higher individual profit than in a strong and weak federation, but the total profit of federation decreases. However, an extremely high selfishness level may deter the generation of additional individual profit.

In the present work, the forwarding policy of each CSP is consider as its strategic leverage. We plan to extend our work by investigating federation modes where the strategies of CSPs will be expressed through both forwarding policy and pricing. We also plan to study different types of federation based on an alternative model, where the federation is instantiated through computational capacity sharing instead of sharing tasks.

Acknowledgment

This work was partly funded by the Research Centre of Athens University of Economics and Business, in the framework of the project entitled “Original Scientific Publications”, and by the EU Project SmartenIT (FP7-2012-ICT-317846).

References

[1] http://onapp.com/federation.
[2] http://www.arjuna.com/federation.
[3] https://www.egi.eu/infrastructure/cloud.
[4] http://www.bonfire-project.eu.
[5] http://openlab.web.cern.ch.
[6] L. Mashayekhy, M. Nejad, and D. Grosu, “Cloud federations in the sky: Formation game and mechanism,” *IEEE Trans. on Cloud Computing*, vol. 3, no. 1, pp. 14–27, Jan 2015.
[7] M. Guazzone, C. Anglano, R. Aringhieri, and M. Sereno, “Distributed coalition formation in energy-aware cloud federations: A game-theoretic approach (extended version),” *CoRR*, vol. abs/1309.2444, 2013.
[8] I. Goiri, J. Guitart, and J. Torres, “Characterizing cloud federation for enhancing providers’ profit,” in *Proc. of IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 2010.
[9] H. Li, C. Wu, Z. Li, and F. Lau, “Profit-maximizing virtual machine trading in a federation of selfish clouds,” in *Proc. of IEEE INFOCOM*, 2013.
[10] N. Samaan, “A novel economic sharing model in a federation of selfish cloud providers,” *IEEE Trans. on Parallel and Distributed Systems,*, vol. 25, no. 1, pp. 12–21, Jan 2014.
[11] G. Darzanos, I. Koutsopoulos, and G. D. Stamoulis, “A model for evaluating the economics of cloud federation,” in *Proc. of 4th IEEE International Conference on Cloud Networking*, 2015.
[12] R. Buyya, R. Ranjan, and R. N. Calheiros, “Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services,” in *Proc. of the 10th International Conference on Algorithms and Architectures for Parallel Processing - Volume Part I*, 2010.
[13] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Muoz, and G. Tofetti, “Reservoir - when one cloud is not enough,” *Computer*, vol. 44, no. 3, pp. 44–51, March 2011.
[14] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, “Cloud federation,” in *Proc. of Cloud Computing*, 2011.
[15] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. M. Sadjadi, and M. Parashar, “Cloud federation in a layered service model,” *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1330–1344, 2012.
[16] M. Hassan, B. Song, and E.-N. Huh, “Distributed resource allocation games in horizontal dynamic cloud federation platform,” in *Proc. of IEEE 13th International Conference on High Performance Computing and Communications (HPCC)*, 2011.
[17] Y. Wang, X. Lin, and M. Pedram, “A game theoretic framework of sla-based resource allocation for competitive cloud service providers,” in *Proc. of Sixth Annual IEEE Green Technologies Conference*, 2014.
[18] M. Steinder, I. Whalley, J. Hanson, and J. Kephart, “Coordinated management of power usage and runtime performance,” in *Proc. Network Operations and Management Symposium (NOMS 08)*, 2008.
[19] L. S. Shapley, “A value for n-person games,” Tech. Rep., 1952.
[20] G. Debreu, “A social equilibrium existence theorem,” *Proc. of the National Academy of Sciences of the United States of America*, 1952.