University of Zurich[UZH]

Patrick Poullie and Burkhard Stiller

Cloud Flat Rates Enabled via Fair Multi-resource Consumption

TECHNICAL REPORT – No. IFI-2015.03

October 2015

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

ifi

Cloud Flat Rates Enabled via Fair Multi-resource Consumption

Patrick Poullie, Burkhard Stiller
University of Zürich, Department of Informatics (IFI), Communication Systems Group (CSG),
Binzmühlestrasse 14, CH–8050 Zürich, Switzerland
poullie@ifi.uzh.ch, stiller@ifi.uzh.ch

***Abstract*—Many companies rent Virtual Machines (VM) from cloud providers to meet their computational needs. While this option is also available to end-users, they take advantage of this option much less. One reason may be that it is common to pay on a per-VM-basis, whereas the telecommunications sector has shown that customers prefer flat rates. A flat rate for cloud services needs to define utilization thresholds, to cap the usage of heavy customers and thereby limit their impact on the flat rate price and the cloud performance. Unfortunately, customers consume multiple heterogenous resources in clouds, *e.g.*, CPU, RAM, disk I/O, or network access. This makes the definition of a customer's fair "cloud share" and according utilization thresholds complex.**

This paper defines fair multi-resource cloud sharing and details its enforcement to enable attractive cloud sharing schemes that, particularly, enable cloud flat rates. Based on a questionnaire among more than 600 individuals, a new Greediness Metric (GM) is designed, which formalizes an intuitive understanding of multi-resource fairness without access to consumers' utility functions. By simulating the GM's application to adapt resource allocations of cloud hosts, it is shown how fairness between cloud customers can be achieved and thereby attractive cloud flat rates offered.

I. Introduction

Cloud Computing (CC) is a computing paradigm enabled by the growing connectivity provided by modern communication systems combined with virtualization technology [25]. CC allows server farms to provide their combined computing power on demand to customers, such as end-users and companies. To process a workload through CC a customer starts Virtual Machines (VM) in the cloud that process the workload. In particular, a cloud processes varying workloads efficiently by dynamically starting VMs, which are defined by Virtual Resources (VR), *e.g.*, virtual CPU (VCPU) and virtual RAM (VRAM).

Resources in private clouds are often managed by quotas, *i.e.*, each user has a quota defining a maximum of VRs that his VMs may have in total. As opposed to this, in commercial clouds, it is common practice that customers pay on a per-VM-basis (different prices for different VM configurations apply). However, the telecommunications sector has shown that customers often prefer flat rates [1], [20], [21], [22], even if a volume-based tariff would reduce costs [17]. With a cloud flat rate customers pay a monthly fee to get access to a cloud, where they can start VMs. Just as Internet flat rates come with a maximum bandwidth, cloud flat rate customers would get a certain quota to spawn VMs from. However, this mechanism is neither sufficient to cap costs that individual customers cause nor to ensure fairness between customers, because (i) customers often deploy different amounts of their quota and (ii) VMs may have very different loads. In particular, (i) means that some customers may only operate a few VMs leaving most of their quota unused, while others fully utilize their quota. However, (ii) describes the more important fact that customers may burden their VMs with very different workloads. That is, even if customers create exactly the same number and types of VMs, the costs they cause varies depending on how they utilize their VMs. This becomes even more relevant, because VMs on the same host compete for resources, *i.e.*, a heavily loaded VM may impair the performance of other VMs on the same host. Therefore, when clouds are shared via flat rates, it is desirable to limit VMs of heavy customers in favor of VMs of more moderate customers, such that VMs of moderate customers are not impaired by VMs of heavy customers. While technical means to enforce fairness in this way exist [2], [11], [13], cloud fairness is neither sufficiently enforced nor explored [7], [10], [15], [16], [26]. Here, *multi-resource* implies that the *bundles*, which are allocated to the consumers, consist of multiple heterogenous resources and the non-accessibility of utility functions implies that the definition can not be based on how much consumers valuate different bundles (a consumer's *utility function* maps each bundle to a number that quantifies the consumer's valuation for the bundle).

The problem of cloud fairness is often addressed by VM scheduling [3], [9], *i.e.*, deciding which VM should be started next, which is insufficient to streamline actual resource utilization during runtime. Other approaches make assumptions about the utility functions of VMs [7], [10], [26].

This paper, therefore, defines the Greediness Metric (GM) to quantify the commensurability of VMs' runtime resource utilization, *i.e.*, how fair VMs behave. Because fairness is an intuitive concept, that is, differs from person to person, the conformance of the GM with an intuitive understanding of fairness is verified by a questionnaire among more than 600 participants. The GM is applied on each host of a cloud to quantify the *greediness* of each VM. This greediness of VMs is inquired from each host and the greediness of each customer's VMs are summed together. These sums are returned to all hosts, such that hosts can constrain VMs of customers with a high sum, if congestion on any resource occurs. In this way, VM greediness is aggregated to customer greediness, such that

local (per host) resource allocations to VMs can be adapted to ensure global (between customers) fairness.

The remainder of this paper is structured as follows: Section II discusses related work leading to the research question stated in Section III. Section IV outlines the questionnaire, presents its results, and discusses key findings. Based on these outcomes, the new Greediness Metric (GM) is defined in Section V and evaluated in Section VI. Section VII summarizes key insights gained and draws conclusions.

II. Related Work

Fairness is an illusive concept, *i.e.*, differs form person to person. However, for a single resource, the size of bundles can be quantified, and thus, the value of bundles objectively compared. Therefore, fairness can be intuitively defined as Max-Min-Fairness [24] and quantified by metrics such as [14], [18], [28]. While such allocation problems were extensively studied in computer science [5], [24], the multi-resource allocation received much less attention by computer scientists. In economics, fairness of multi-resource allocations is defined by envy-freeness [4], *i.e.*, no consumer prefers to swap his bundle with another. However, such definition is only applicable, when consumer's utility functions are known.

As noted in [7], [10], [15], [16], [23], [26], multi-resource allocation in data centers is so far not fully investigated and often reduced to single-resource allocation problems at the cost of efficiency and fairness [10], [19]. In data centers consumers share resources, such as CPU time, RAM, Disk I/O and space, and network access, wherefore it is necessary to define fairness when every consumer receives a bundle of multiple heterogenous resources. Clouds are a special case of data centers where resources are deployed via VMs in a highly dynamic manner. The assumption in literature is that the resource utilization of jobs, which are the entities consuming resources in data centers, is static [7], [10]. Accordingly, approaches to fairness in data centers focus on scheduling, *i.e.*, which job should be started next. However, because VMs change their resource utilization, resource allocation in clouds is more complex and cannot be reduced to VM scheduling. Subsequently, approaches to multi-resource fairness in data centers (especially their cloud instances) are compared, while ignoring numerous approaches focusing on a single resource. Table II shows this overview of approaches by comparing them in terms of the adopted fairness definition, the application area, and the assumed utility function.

Dominant Resource Fairness (DRF) is the most prominent approach to introduce multi-resource fairness in data centers [9]. DRF defines the value of a bundle as the biggest proportion relative to the total supply of any resource in it (*c.f.* Section IV-C1d). Therefore, to define the value of a bundle only one resource is considered, which is also known as the L_∞ norm. A DRF fair allocation is the allocation that maximizes this value for every consumer. [12] points out that for many other functions (including all other $L_{i\in\mathbb{N}}$ norms) a unique allocation exists that can be found in polynomial time, but that the authors of DRF never argue why their choice is superior. Leontief utility functions model that resources are required in static ratios, *i.e.*, increasing the amount a consumer receives of one resource does not increase his utility if his share of all other resources is not increased by the same factor. While all proofs for DRF's desirable properties are based on Leontief utility functions [9], [10], [23], the actual DRF scheduling algorithm, as proposed in [9], [10], allows consumers to request different resource bundles. Therefore, in the scheduling process, consumers can have arbitrary utility functions, but DRF's properties are only proven for Leontief utility functions. [27] points out that in practice DRF may not allocate desired resources to a consumer, although it would not hurt other consumers. [23] justifies that DRF may perform poor with respect to overall system utilization/social welfare by impossibility results for sharing policies with DRF's properties. However, [3] shows that proportional fairness is superior to DRF in terms of efficiency and comes with the same desirable characteristics under realistic assumptions. In particular, [3] argues that DRF's desirable properties are proven for simplistic micro-economic assumptions. [12] extends DRF to perfectly complementary utility functions and allows all monotonic and continuous functions to measure the value of bundles. While [12] shows that for this generalization a unique allocation exists, which can also be found in polynomial time, it remains open whether this approach is strategy-proof or envy-free.

TABLE I: Comparison of related approaches.

App.	Fairness	Area	Utility Function
DRF	Max-min for L_∞ norm	Scheduling	Leontief
[12]	Max-min for monotonic continous norm	General	Perfectly complementary
BBF	At least equal share on a bottleneck resource	Scheduling	Perfectly complementary
[26]	Envy free and Pareto efficient	Micro resources	Cobb-Douglas
[6]	Proportional fairness	Theory	Homogeneous of degree one
[19]	Fair on progress shares	Scheduling	by profiling/static
[25]	Not defined	Scheduling	Upon requests
[16]	Priority functions	Runtime	Not needed
GM	Based on questionnaire	Runtime	Not needed

Bottleneck-based fairness (BBF) was introduced in [7], [8]. An allocation is bottleneck-based fair, if every consumer either is allocated all requested resources or at least the equal share on a fully utilized/bottleneck resource and the other resource in proportion to this bottleneck. [27] defines a multi-resource on-line scheduling policy that achieves BBF without knowing consumer's utility functions in advance.

[26] presents an allocation policy that achieves game-theoretic fairness, *i.e.*, sharing incentive, envy freeness, and pareto-efficiency, when allocating cache capacity and memory bandwidth. [26] profiles different applications to convincingly argue that Cobb-douglas utility functions are well suited to model diminishing returns and substitution effects, when cache capacity and memory bandwidth are consumed. The authors note that also other microarchitectural cloud resources, *e.g.*, number of processor cores, could be allocated by their mechanism.

The results in [6] are all theoretic in nature. A mechanism is presented that may hold resources back, to incentivize con-

sumers to be truthful and, thereby, ensure strategy-proofness. This behavior is theoretically justified by showing that no strategy-prove mechanism can guarantee to every consumer a utility greater than 0.5 of the proportional fair utility. However, it makes the mechanism unattractive to be deployed in practice due to its artificially decreased efficiency.

[19] considers fairness of progress shares, which "capture the contribution of each resource to the progress of a job". Since the special case of data analytics in the cloud is considered, it is possible to assume a constant VM demand, which is not the case in general.

Although [25] considers the realistic case where consumers may lie about their utility functions over multiple resources, it is not elaborated how fairness is defined.

[16] considers the fair sharing of grids between customers and research groups. It points out that clusters, which provide several resources, such as CPU time, RAM, GPUs, and HDD storage, are often shared based on priorities, which solely depend on utilized CPU time. They point to situations, where a customer utilizes 100% of a machine's RAM, but only 10% CPU. This makes the remaining 90% of CPU unusable, while the customer is only accounted for a 10% usage. Contrary to all other approaches discussed in this section, [16] achieves fairness by a penalty/priority function, which is close to our approach. In particular, such function determines the priority of consumers when granting resource requests. Such approach is much more practically oriented, because no assumptions about utility functions need to be made. [16] is dedicated to finding a penalty function that suffices five requirements and compares different penalty functions with respect to these requirements. [16] proves that three of these requirements cannot be achieved simultaneously. Furthermore, the sum-based-penalty function that suffices all but one requirement and the root-based-penalty function that suffices all but another requirement are presented.

III. Problem Statement

While fairness issues arise in many areas of (human) coexistence, for example, when asking the question of how a governing coalition shall be formed in a parliamentary system or how this coalition, once formed, shall allocate cabinet ministries [4], the allocation of resources is probably the most basic context in which fairness issues arise.

The characteristic of fair cloud resource allocations, which is distinct for this context, is that customers (between whom fairness is to be defined) utilize resources from different resource pools (hosts) by intermediaries (VMs). While fairness has to be achieved between customers, this has to be done by allocating resources to their VMs. Unfortunately resources can only be moved between VMs, which run on the same host. However, besides this structural problem, also a more general problem is faced: Defining and enforcing fairness of multi-resource allocations without knowing consumers' utility functions. In particular, because bundles consist of multiple heterogenous resources, bundles can contain resources in different amounts, This prohibits an objective comparison of bundles. For example, some customers may require more CPU for their workloads, while others require more RAM [19]. A third customer may deploy the cloud for backups and, therefore, mostly requires disk-space and bandwidth. Therefore, consumers can have different preferences over the same bundles.

Because utility functions of consumers in clouds are unknown (cf. Section II), it is not possible to define fairness via utility functions here and, thus, fairness has to be defined via bundles that VMs serve themselves. In particular, hosts work as "self-serving stores" for VMs, which means that they provide all requested resources to a VM, if possible. Therefore, fairness in clouds has to be defined as constraining those VMs that overcharge their self-serving store, *i.e.*, are greedy. While also the concept of greediness has no formal definition, it can be better defined and quantified with the information that is available in clouds.

Therefore, the ***problem statement*** for this paper reads as follows: The greediness of consumers can be defined and quantified based on their multi-resource self-servings. An allocation is fair in such a case, when (i) this quantification is aligned for all consumers and (ii) greedy consumers are constrained in favor of less greedy consumers.

IV. Questionnaire

Fairness and greediness are intuitive concepts, *i.e.*, they differ in their perceptions from person to person. Thus, a questionnaire was developed to evaluate their intuitive understanding and to justify empirically the design of this new Greediness Metric (GM). DRF's conformance with an intuitive understanding of fairness was also evaluated in this questionnaire, since (i) DRF is the most prominent approach for fairness in data centers and (ii) respective replies serve as checks for questionnaire reply qualities. Therefore, this questionnaire allows to derive a justified and intuitive understanding of fairness and greediness, for the case where little or no information about consumers' utility functions is available. As discussed above, avoiding any explicit information on utility functions was decided on purpose, because in case of clouds, utility functions are not known.

The questionnaire specified real-life scenarios in terms of three questions Q1, Q2, and Q3 to not distract participants by technical terms and let them fully concentrate on the question of fairness. For example, Q3 describes a scenario, where three bakers purchase together three ingredients for cakes. The bakers prepare different recipes and split the ingredients by putting them on a table from which every baker serves himself. This was done until at least one resource on the table was depleted, which prohibited the utilization of the other resources. While this is a specific scenario the underlying problem is generic: Common heterogenous resources are split among individuals with different demands by letting them serve themselves. This implies mutual trust and poses the question of how individuals, who could try to exploit the system, can be identified, that is, how disproportionate consumption can be defined. The transferability to clouds is evident: just as the bakers, VMs serve themselves from a pool of common resources (the resources of their host). Defining disproportionate consumption is necessary to prioritize moderate VMs in case of congestion.

Questionnaire participants had to chose between different options of allocations or define rankings of consumers. Additionally, participants were offered to explain their answers in text boxes. The questionnaire did not address any particular target group. Out of 721 participants, who started the questionnaire, 604 completed it.

A. Choosing the Most Fair Allocation (Q1)

Q1 was designed to evaluate a key building block of DRF: the use of the L_∞ norm to measure the value of a bundle, *i.e.*, that the value of a bundle is defined solely by the resource that it contains in the largest proportion (*c.f.* Section IV-C1d). The scenario described covers two resources r_1 and r_2 of which six and twelve units where available, respectively These resources have to be allocated to three consumers c_1, c_2, and c_3. c_1 only requires r_1, c_2 only r_2, and c_3 requires for each unit of r_1 two units of r_2. This results in seven possible allocations to allocate all resources and do not give consumers resources they have no use for. However, most of these allocations are intuitively unfair, *e.g.*, in two of these allocations at least one consumer receives no resources at all. Because the scenario describes that resources are requested in static ratios, it is transferable to allocation problems in data centers, where these static ratios of resource requests are the standard assumption [7], [10].

Participants were presented with four of the seven possible allocations (namely as listed in Table II) and asked to choose *the* allocation that seemed most fair to them. The allocations to choose from were presented numerically and graphically to participants (cf. Appendix A). As mentioned, Table II shows the allocations participants had to choose from and the respective labels (allocation A11, A12, A13, and A14). As expected, A11 and A14 were only chosen by a minority of the 721 participants (0.4% and 1.1%, respectively) and most participants deterred between A12 and A13 (30.0% and 68.5%, respectively). The following arguments in support of A12 are summarized from textual comments received:

- c_1 and c_2 only compete with c_3 for resources, but not with one another. A fair allocation splits resources equally between those who contend for them.
- All receive an equal amount of what they want.
- This is the only allocation where nobody can complain that someone has more of the same resource.
- All consumers have equal utilities.
- When prices are introduced based on available units, this option gives the same value to all consumers.

For the following reasons A13 was supported:

- When prices are introduced based on available units, this option gives the same value to all consumers.
- c_1 and c_2 receive $2/3$ of *one* resource and c_3 $1/3$ of *two* resources, which makes $2/3$ for everybody. On a similar note, some participants rejected A12, because c_3 gets as much as c_1 and c_2 combined.
- Because c_1 and c_2 only want one resource, they should get more of it than c_3, as c_3 wants both resources.
- This option is the result of a simple auction or when all consumers get an equal share of both resources and then trade.

TABLE II: The four options A11, A12, A13, and A14 in Q1 of the questionnaire to allocate the two resources r_1 and r_2 to the three consumers c_1, c_2, and c_3.

	A11		A12		A13		A14	
Consumer	r_1	r_2	r_1	r_2	r_1	r_2	r_1	r_2
c_1	2	0	3	0	4	0	5	0
c_2	0	4	0	6	0	8	0	10
c_3	4	8	3	6	2	4	1	2

- The range of numbers of units given to consumers is the smallest. In particular, for A13 the range is 4-6-8, while it is 3-6-9 for A12.

B. Allocating Based on Resource Requests (Q2)

Q2 was designed to validate how participants allocate scarce resources based on requests. This problem is relevant in clouds, because, when a host experiences congestion, it has to decide which requests to accept and which to delay or reject. In Q2 three resources r_1, r_2, and r_3 (for each eight units were available) had to be split between the two consumers c_1 and c_2 (cf. Appendix B). All three resources were initially contended, but the contention for r_1 can be resolved in bilateral negotiations, with c_1 receiving two units and c_2 six. However, these negotiations left open how r_2 and r_3 had to be allocated. c_1 demanded five units of both resources and c_2 demanded four units of both. Participants had to decide, whether one consumer has to cease both missing units or each consumer should cease one. Accordingly, the most fair option had to be chosen from allocations A21, A22, and A23 (cf. Table III).

715 participants answered this question. From the information given by participants in the text boxes, the following frequent arguments were compiled, indicating the subjective perception of fairness.

A21 was supported by 13.9% of the participants with these arguments:

- Since r_1 is not under contention anymore, it does not need to be considered for the allocation of r_2 and r_3. Consequently, c_1 and c_2 should get a fair, that is, equal, amount of r_2 and r_3.
- Because c_2 receives more of the first resource, c_2 will probably also require more of the other resources, and thus should not be constrained on these.
- The consumer, who demands the greater amount of a contented resource, should cease the deficit, because relative to his demand this deficit is smaller than for someone who demands less.

A22 was supported by 23.7% of the participants with this argument:

- Because c_1 gets 4 units less of the first resource compared to c_2, c_1 should get four units more of the other resources.

62.4% of the participants supported A23 with this argument:

- Since r_1 is no more under contention, it does not need to be considered for the allocation of r_2. Assuming that the demands reflect actual needs (rather than negotiating positions), the most fair thing to do is splitting the scarcity equally, *i.e.*, both get one resource less than requested.

TABLE III: The three options A21, A22, and A23 in Q2 of the questionnaire to allocate the three resources r_1, r_2 and r_3 to the two consumers c_1 and c_2.

	A21			A22			A23		
Consumer	r_1	r_2	r_3	r_1	r_2	r_3	r_1	r_2	r_3
c_1	2	4	4	2	5	5	2	4	5
c_2	6	4	4	6	3	3	6	4	3

C. Estimating Greediness (Q3)

Q3 as defined in was designed to collect information on how the greediness of consumers, who served themselves from a pool of common resources, is perceived. In addition, insights are collected on how proportionality and value of resource bundles is perceived, when no information about consumers' utility functions is available. Thus, Q3 is most important for the investigation of an intuitive fairness understanding, as it provides insights on how resources that different VMs on the same host utilize can be compared.

Q3—as defined in Appendix C—is based on three scenarios S31, S32, and S33, were three consumers c_1, c_2, and c_3 had served themselves from a pool of three common resources r_1, r_2, and r_3 (like VMs on the same host). To split these resources, each consumer had allocated himself a certain bundle as shown in Table IV. The three consumers had to be ranked according to how their greediness was perceived, all being based on the amounts the consumers had allocated themselves.

1) Metrics: Many participants tackled Q3 by proposing a metric to assess the value of bundles. These metrics encompasses the following four: price, price×scarcity, price∩scarcity, and DRF.

a) Price: The price metric is the simplest metric and was often suggested by participants. The value of one unit of resource r_i is defined as $p/\overleftrightarrow{r_i}$, where p is a constant and $\overleftrightarrow{r_i}$ is the number of units available of r_i. The value of a bundle is the sum of values of its resources. For example, for $p = 1$ the value of c_2's bundle in S31 is

$$\frac{2}{12} + \frac{1}{9} + \frac{5}{9} = \frac{5}{6}.$$

This metric is equivalent to the sum-based-penalty function presented in [16].

b) Price×scarcity (P×S): The price×scarcity metric is a natural extension of the price metric. The value of one unit of resource r_i is defined as $a(r_i) \cdot p/\overleftrightarrow{r_i}^2$, where $a(r_i)$ is the amount that is allocated in total of r_i. The value of a bundle is defined as the sum of values of those resources contained. For example, for $p = 1$ the value of c_2's bundle in S31 is

$$\frac{2 \cdot 10}{12^2} + \frac{1 \cdot 6}{9^2} + \frac{5 \cdot 9}{9^2} = \frac{29}{54}.$$

c) Price∩scarcity (P∩S): The price∩scarcity metric is another natural extension of the price metric and defines the value of a resource just as the price metric. However, the value of a bundle is defined only over resources that are depleted, *i.e.*, resources where $a(r_i) = \overleftrightarrow{r_i}$. For example, for $p = 1$ the value of c_2's bundle in S31 is

$$\left\lfloor \frac{10}{12} \right\rfloor \cdot \frac{2}{12} + \left\lfloor \frac{6}{9} \right\rfloor \cdot \frac{1}{9} + \left\lfloor \frac{9}{9} \right\rfloor \cdot \frac{5}{9} = \frac{5}{9}.$$

TABLE IV: The three scenarios S31, S32, and S33 in Q3 of the questionnaire, where three consumers c_1, c_2, and c_3 served themselves form a pool of three common resources r_1, r_2 and r_3.

	S31			S32			S33		
Consumer	r_1	r_2	r_3	r_1	r_2	r_3	r_1	r_2	r_3
c_1	4	3	3	4	2	4	4	1	4
c_2	2	1	5	1	4	3	1	4	3
c_3	4	2	1	1	6	2	1	6	2
Remainder	2	3	0	6	0	0	6	1	0

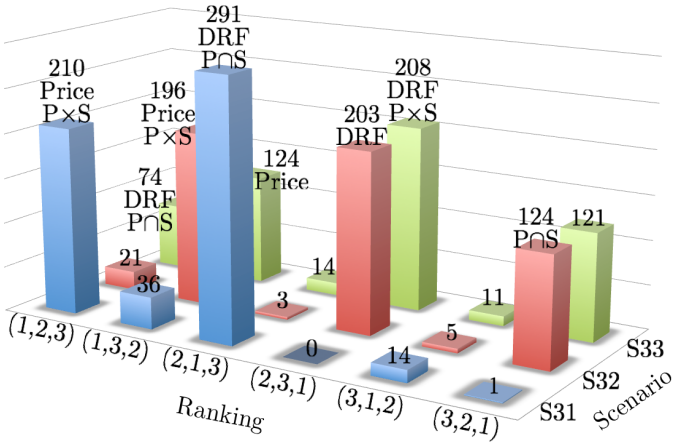


Fig. 1: Number of participants who selected the different rankings (represented by triplets) in the three scenarios of Q3.

d) DRF: DRF (Dominant Resource Fairness) defines the value of a bundle by the L_∞ norm, *i.e.*, by the biggest share of any resource relative to the overall amount of the resource. For example, the value of c_2's bundle in S31 is:

$$\max\left(\frac{2}{12}, \frac{1}{9}, \frac{5}{9}\right) = \frac{5}{9}.$$

According to this DRF metric, the bundles of c_1 and c_2 in S33 are equally valuable. This tie is broken by considering the second biggest share in the bundles, wherefore c_1's is more valuable.

2) Frequency Investigations: In this section numerical results of Q3 are discussed. For some rankings the free text

TABLE V: Percentages of most frequent rankings in Q3.

Metric	S31	S32	S33
DRF	(2,1,3): 52.7%	(2,3,1): 36.8%	(2,3,1): 37.7%
Price	(1,2,3): 38.0%	(1,3,2): 35.5%	(1,3,2): 22.5%
P×S	(1,2,3): 38.0%	(1,3,2): 35.5%	(2,3,1): 37.7%
P∩S	(2,1,3): 52.7%	(3,2,1): 22.5%	(1,2,3): 13.4%

TABLE VI: Quantity of participants who selected the most frequently selected combinations of rankings in Q3.

Quantity	S31	S32	S33	Conforming with
79	(2,1,3)	(2,3,1)	(2,3,1)	DRF
55	(1,2,3)	(1,3,2)	(2,3,1)	P×S
43	(1,2,3)	(1,3,2)	(1,3,2)	Price
32	(2,1,3)	(3,2,1)	(1,2,3)	P∩S
30	(2,1,3)	(1,3,2)	(1,3,2)	
28	(2,1,3)	(3,2,1)	(3,2,1)	
27	(1,2,3)	(3,2,1)	(3,2,1)	
24	(2,1,3)	(2,3,1)	(3,2,1)	
22	(1,2,3)	(2,3,1)	(2,3,1)	
20	(1,2,3)	(2,3,1)	(3,2,1)	
18	(2,1,3)	(2,3,1)	(1,3,2)	
17	(2,1,3)	(1,3,2)	(2,3,1)	
16	(2,1,3)	(2,3,1)	(1,2,3)	
12	(1,3,2)	(1,3,2)	(1,3,2)	
9	(2,1,3)	(1,3,2)	(1,2,3)	
8	(2,1,3)	(3,2,1)	(2,3,1)	

indicated that the participant had assumed real life values of the resources and ranked the consumers accordingly. These rankings, as well as, incomplete rankings were removed, wherefore the presented results are based on 553 answers.

Subsequently, consumer rankings are denoted by triplets. For example, when the first consumer is ranked moderate, the second consumer is ranked most greedy, and the third consumer is ranked least greedy, this is denoted by the triplet (2,1,3).

Figure 1 illustrates for each scenario how many participants selected each ranking and highlights the rankings that correspond to the metrics discussed in Section IV-C1. Table V shows which ranking corresponds to which metric for the three scenarios. Furthermore, Table V provides the percentage of participants, that answered with the respective ranking (deduced directly form numbers of Figure 1).

Because these rankings given by most participants do not match the same metric over the three scenarios, Table VI shows the quantity of the sixteen most prominent combinations of rankings (which cover 79.6% of all participants). In addition to these combinations, three combinations were selected five and seven times each, two combinations were selected four and six times each, 4 combinations were selected three times, and 15 combinations were selected one and two times each.

D. Discussion

1) Choosing the Most Fair Allocation (Q1): Only the first two arguments in favor of A12 are correct. The third argument (nobody can complain that someone has more of the same resource) is incorrect, because c_1 receives least of r_2 and c_2 least r_1. The fourth argument (all consumers have equal utilities) lacks foundation, because nothing was said about utilities of consumers, only about ratios in which they request resources. The fifth argument states that all receive the same value, which is actually the case for A13. Therefore, the arguments given in text boxes in favor of A13 are more versatile and credible than those in favor of A12. Because A12 and A13 were chosen by 30.0% and 68.5% of the participants, respectively, that is, A13 was chosen more than 2.25 times as often as A12, it is concluded that A13 determines the intuitively fair allocation.

2) Allocating based on Requests (Q2): While supporters of A21 and A23 stated that the allocation of r_1 does not need to be considered, when allocating r_2 and r_3, both concluded differently on how r_2 and r_3 should be allocated: supporters of A21 argued that it is fair to *allocate the resources fair*, *i.e.*, give both an equal amount of the resources, while supporters of A23 argued that it is fair to *allocate the deficit fair*, *i.e.*, give each consumer one resource less, although this means that c_2 receives two more additional. Allocating the deficit in a fair manner, was frequently justified by the assumption that demands reflect actual needs. In general, participants considered the information important, which consumer ceased more demands for r_1 and whether demands reflect actual needs or negotiation positions. Although A22 is most fair according to all metrics discussed in Section IV-C1, only 23.7% of participants selected it. In contrast, A23 was selected by 62.4% of participants, that is, almost three times as often. This shows that most participants consider it fair to allocate the deficit fairly and not the actual resources. Unfortunately, giving a larger amount of resources to consumers, who demand more, will give incentives to VMs to exaggerate demands, which is already sometimes the case [10].

3) Estimating Greediness (Q3): Many participants stated to have struggled ranking the consumers, because no information was given about how efficiently these consumers used their resources. Unfortunately, such uncertainties are also present in clouds, where the performance of VMs can only in some cases be related to their resource consumption.

Numbers of Table V show that these rankings according to the four metrics presented in Section IV-C1 cover 90.7% (S31), 94.8% (S32), and 73.6% (S33) of the participants' rankings. Despite this high coverage it was often the case that a participant's ranking was conforming with different metrics over the three scenarios. The four most prominent combinations ((2,1,3), (2,3,1), (2,3,1)), ((1,2,3), (1,3,2), (2,3,1)), ((1,2,3), (1,3,2), (1,3,2)), and ((2,1,3), (3,2,1), (1,2,3)) are conforming with DRF, P×S, Price, P∩S, respectively. However, as discussed next, the two top combinations were often did not result from applying the respective metric.

The most prominent combination was chosen by 79 participants. Although this combination results from the DRF metric, only one participant argued according to DRF. Interestingly, participants who opted for this combination provided more often explanations than those who opted for other combinations. The most frequent argument was, that those who exceed their equal share of a resource are considered greedy. Thus, in S31 c_2 is the greediest due to the disproportional consumption of r_3. Analog, in S32 and S33 c_3 is the greediest, because c_3 exceeds the equal share of r_2 by 50% and c_1 is the second greedy, because c_1 exceeds the equal share of r_1 by 33%. The combination of rankings chosen by 55 participants is conforming with the P×S metric. Applying this metric implies several calculations, but participants arrived at this ordering differently, as inferred from their comments: most ordered the consumers according

to the overall amount they had used and broke the tie of c_1 and c_3 in S33 by c_3's 50% overconsumption of r_2. The combinations of rankings chosen by 43 and 32 participants are conforming with the price and P∩S metric, respectively. Most participants stated that they had applied these metrics to arrive at the ranking. Note that, for the combination of the P∩S metric, the ranking from S32 to S33 is inverted, where in S32 c_1 was least greedy and c_3 most greedy. Two participants already noted in their comments that this reordering is counter-intuitive, because c_1 actually takes up less resources in S33, while consumptions of the other two consumers are stable.

4) Implications for existing metrics: Although the metrics discussed in Section IV-C1 cover the majority of participants rankings, none of these metrics captures an intuitive understanding of fairness: The P∩S metric has a low conformance in S32 and S33 and for S33 results in the inverted ranking of S32, which means that consumers can decrease their ranking by adding additional resources to their bundle. This not only is counter-intuitive, but also gives incentives to consumers to consume more resources than needed. The Price metric has an insufficient conformance in S33 and identifies c_1 as "greediest" in S31, although c_1 does not cause the bottleneck, but precisely sticks to his equal share (a behavior that is considered humble in S32 and S33). The sum- and the root-based-penalty function presented in [16] result in the same rankings as the Price metric and, therefore, are also not satisfactory. Similar arguments can be made for the P×S metric.

The DRF metric is satisfactory at a first glance: For all three scenarios of Q3 it results in the most frequent ranking and also in the most frequent combination of rankings (cf. first body row of Table V and Table VI). However, only one participant argued according to DRF, while the majority argued that those who exceed their equal share are greedy. Therefore, the high conformance of DRF stems rather from the fact that every consumer exceeds his equal share on at most one resource, which allows DRF to produce good results, because it only considers these resources. To show that DRF's approach to ignore all but one resource can lead to undesirable results, a sample allocation is presented in Table VII. For this allocation all other metrics discussed in Section IV-C1 give the inverse ranking of DRF. Also, according to the arguments made by the participants, the DRF ranking is unfair: DRF classifies c_1 as the consumer with the most valuable bundle, although c_1 only receives the least loaded resource. c_3 cedes no resource at all and receives most of the only scarce resource, but DRF classifies c_3 as most humble.

Q1 of the questionnaire identified A13 as the intuitively fair allocation, while the DRF-fair allocation is A12. Because Leontief utility functions are assumed in Q1, *i.e.*, resources are required in static ratios, and DRF is defined based on this assumption, DRF should result in an intuitively fair allocation, when this assumption holds. Therefore, while DRF is often applied, when Leontief utility functions do not hold, Q1 shows that already for Leontief utility functions, DRF may result in allocations that are not intuitively fair. Moreover, consumers and resources can be added to this scenario, where consumer c_i requests only resource r_i (and as before one consumer requests all resources evenly). Thereby, the perceived unfairness of DRF can be increased arbitrarily, because the consumer requesting all resources would receive as much as all other consumers combined. In other words, DRF-fair allocations exist with a higher degree of intuitive unfairness.

TABLE VII: An example of a problematic DRF ranking.

	r_1	r_2	r_3	Price	P×S	P∩S	DRF
available	30	30	30				
c_1	18	0	0	1.80	1.68	0.00	0.60
c_2	0	14	17	3.10	3.04	1.40	0.56
c_3	10	16	12	3.80	3.69	1.60	0.53

Due to these shortcomings of all metrics hitherto presented, a new Greediness Metric (GM) is developed and introduced. This GM is aligned with the arguments of participants and accordingly (i) results in the most frequent rankings for each of the three scenarios in Q3, (ii) classifies A13 as the most fair allocation, and (iii) gives the "correct" ranking for the allocation in Table VII.

V. Greediness Metric

The Greediness Metric (GM) maps each resource bundle in an allocation to a rational number that captures its value and, therefore, can be associated to the *greediness* of the consumer, who served himself the bundle. In that sense this GM serves the same purpose as the other metrics presented in Section IV-C1. The GM sums up, what exceeds the equal share in each bundle, according to the most frequent questionnaire's responses. However, it also deducts what is not consumed of the equal share and is handed over to other consumers instead.

Let $R = (r_1, r_2, \ldots, r_m)$ be a set of m resources, where resource $r_i \in R$ is available in the amount of $\overleftrightarrow{r_i}$. An *allocation* of R to n consumers $(c_1, c_2, \ldots, c_n)$ can be denoted by a matrix $A \in \mathbb{R}_{\geq 0}^{m \times n}$ with $\sum_{j=1}^{n} a_{ij} \leq \overleftrightarrow{r_i}$, for all $i \in \{1, 2, \ldots, m\}$, where c_j receives amount a_{ij} of r_i. The amount of r_i that c_j receives beyond his equal share is then $a_{ij} - \overleftrightarrow{r_i}/n$ (if the difference is negative, c_j does not utilize its entire equal share of the resource).

If $a_{ij} > \overleftrightarrow{r_i}/n$, consumers other than c_j have to cede some of their equal share of r_i in order to enable c_j's share of r_i. Therefore, the amount that exceeds c_j's equal share is added to the greediness of c_j.

If $a_{ij} = \overleftrightarrow{r_i}/n$, c_j exactly receives his equal share, wherefore it does not change c_j's greediness. In particular, if $a_{ij} = \overleftrightarrow{r_i}/n$ for all $i \in \{1, 2, \ldots, m\}$, c_j's greediness is zero.

If $a_{ij} < \overleftrightarrow{r_i}/n$, c_j's cession of r_i is credited to c_j, *i.e.*, subtracted from c_j's greediness, to the extent that other consumers profit from this cession, which is the case, when they utilize r_i beyond their equal share. This extension not only depends on how much of r_i is utilized beyond the equal share by other consumers, but also on how much of r_i is ceded by other consumers. Therefore, the *credit factor* for the cession of r_i is the ratio of what is ceded of r_i to what is consumed beyond the equal share of r_i. To capture this notion formally,

TABLE VIII: GM results for Q1 and Q3 of the questionnaire.

γ	Cons.	A12	A13	S31	S32 and S33
x	c_1	$0.25-0.5\cdot x$	$0.5-0.5\cdot x$	0	$0.\overline{1}-0.1\overline{6}\cdot x$
	c_2	$0.25-0.5\cdot x$	$0.5-0.5\cdot x$	$0.\overline{2}$	0
	c_3	0.5	0	$-0.\overline{2}\cdot x$	$0.1\overline{6}-0.\overline{1}\cdot x$
1	c_1	-0.25	0	0	$-0.0\overline{5}$
	c_2	-0.25	0	$0.\overline{2}$	0
	c_3	0.5	0	$-0.\overline{2}$	$0.0\overline{5}$
0.5	c_1	0	0.25	0	$0.02\overline{7}$
	c_2	0	0.25	$0.\overline{2}$	0
	c_3	0.5	0	$-0.\overline{1}$	$0.\overline{1}$
0.25	c_1	0.125	0.375	0	$0.069\overline{4}$
	c_2	0.125	0.375	$0.\overline{2}$	0
	c_3	0.5	0	$-0.0\overline{5}$	$0.013\overline{8}$

$\alpha(r_i)$ defines the sum of what consumers receive beyond their equal share of r_i, *i.e.*,

$$\alpha(r_i) := \sum_{j=1}^{n} \max\left(0, a_{ij} - \overleftrightarrow{r_i}/n\right),$$

and $\beta(r_i)$ defines the sum of what consumers cede of r_i, *i.e.*,

$$\beta(r_i) := \sum_{j=1}^{n} \max\left(0, \overleftrightarrow{r_i}/n - a_{ij}\right).$$

Multiplying the amount that c_j cedes of r_i with $\alpha(r_i)/\beta(r_i)$ implements the considerations above. Therefore, the *greediness of* c_j is defined as

$$g(c_j) := \sum_{i=1}^{m} o(i,j) \cdot n/(m \cdot \overleftrightarrow{r_i}), \tag{1}$$

where the factor $n/(m\cdot\overleftrightarrow{r_i})$ normalizes resource units. Section V-C discusses the normalization in detail. The *offset* $o(i,j)$ for c_j's consumption of r_i and defined as

$$o(i,j) := \begin{cases} a_{ij} - \overleftrightarrow{r_i}/n & \text{if} \quad a_{ij} \geq \overleftrightarrow{r_i}/n, \\ \gamma \cdot \frac{\alpha(r_i)}{\beta(r_i)} \cdot \left(a_{ij} - \overleftrightarrow{r_i}/n\right) & \text{else.} \end{cases} \tag{2}$$

Note that, if $\beta(r_i) = 0$, no consumer cedes r_i and, therefore, the else-part of Equation 2 is never reached (and no division by zero occurs). While $\frac{\alpha(r_i)}{\beta(r_i)}$ in the else-part dynamically influences in dependence of the consumption of r_i how much ceding r_i is credited, this crediting is also statically adjusted by Parameter $\gamma \in [0,1]$. Section V-B discusses the choice of γ.

A. GM Examples

This section provides exemplary calculations of the GM for selected allocations of Q1 and Q3 of the questionnaire. Table VIII summarizes the GM results for Q1 and Q3 of the questionnaire.

1) A12: In the scenario of Q1, one unit of r_1 has a normalized value of $n/(m\cdot\overleftrightarrow{r_1}) = 3/(2\cdot 6) = 0.25$ and one unit of r_2 has a normalized value of $n/(m\cdot\overleftrightarrow{r_2}) = 3/(2\cdot 12) = 0.125$. The equal shares are $\overleftrightarrow{r_1}/3 = 6/3 = 2$ and $\overleftrightarrow{r_2}/3 = 12/3 = 4$ for resource r_1 and r_2, respectively. c_1 has a greediness of

$$g(c_1) = \overbrace{(3-2)\cdot 0.25}^{\text{Overcons. of } r_1} + \gamma\cdot\overbrace{\frac{2+2}{4}\cdot(0-4)\cdot 0.125}^{\text{Ceding of } r_2} = 0.25-0.5\cdot\gamma.$$

In particular, the fraction in the second summand results from c_1 ceding four units of r_2 and c_2 and c_3 exceeding their equal share of r_2 by two units each. The subtraction in brackets results because c_1 consumes zero units of r_2, while the equal share is 4.

For c_2 the calculations are inverse, with respect to r_1 and r_2, *i.e.*,

$$g(c_2) = \overbrace{\gamma\cdot\frac{1+1}{2}\cdot(0-2)\cdot 0.25}^{\text{Ceding of } r_1} + \overbrace{(6-4)\cdot 0.125}^{\text{Overcons. of } r_2} = 0.25-0.5\cdot\gamma.$$

Because c_3 does not cede resources, c_3's greediness does not depend on γ:

$$g(c_3) = \overbrace{(3-2)\cdot 0.25}^{\text{Ceding of } r_1} + \overbrace{(6-4)\cdot 0.125}^{\text{Overcons. of } r_2} = 0.5.$$

2) S32 and S33: S32 and S33 of Q3 are similar: the amounts of available resources are the same, resulting in both scenarios in normalized values of $0.08\overline{3}$, $0.08\overline{3}$, and $0.\overline{1}$ and equal shares 4, 4, and 3 for r_1, r_2, and r_3, respectively. Also the allocations are the same, except that c_1 cedes one unit more of r_2 in S33 compared to S32. Because this unit is not utilized by other consumers, this ceding is not credited to c_1, *i.e.*, c_1's greediness does not change. In particular, the greediness of c_1 is calculated as follows:

$$g(c_1) = \overbrace{(4-4)\cdot 0.08\overline{3}}^{\text{Cons. of } r_1} + \overbrace{o(2,1)\cdot 0.08\overline{3}}^{\text{Cons. } r_2} + \overbrace{(4-3)\cdot 0.\overline{1}}^{\text{Cons. of } r_3} = 0.\overline{1}-0.1\overline{6}\cdot\gamma,$$

where in S32

$$o(2,1) = \gamma\cdot\frac{2}{2}(2-4) = -2\cdot\gamma$$

and in S33

$$o(2,1) = \gamma\cdot\frac{2}{3}(1-4) = -2\cdot\gamma.$$

B. Choice of γ

This section discusses the choice of γ to fine-tune the greediness metric to best comply with the questionnaire results. The GM's design is inline with arguments given most frequently by the participants and already without the parameter γ (or, equivalently, $\gamma = 1$), the GM complies with the results of Q1 and S31 of the questionnaire. However, for S32 and S33, the GM results in the ranking (3,2,1) (cf. Table VIII), while the ranking most frequently selected by the participants is (2,3,1) (cf. Table V). This mismatch for $\gamma = 1$ can be explained as follows: In S32 and S33, c_1 exceeds the equal share of r_3 by

33% but also cedes 50% of r_2 to c_3, while c_2 is "neutral". Thus, while c_2's greediness is zero independent of γ, c_1 has a negative greediness for $\gamma = 1$, *i.e.*, for $\gamma = 1$, the GM ranks c_2 greedier than c_1. However, most participants rated c_1 greedier than c_2, because c_1 over-consumes r_3 while c_2 never exceeds the equal share. To account for this argument, γ regulates how strongly the ceding of resources is credited (in addition to the dynamic regulation by $\frac{\alpha(r_i)}{\beta(r_i)}$), where, the smaller γ is chosen, the harder it gets to compensate for exceeding the equal share.

To match the results of Q3, γ has to be chosen, such that ceding 33% of one resource and over-consuming 50% of another resource, results in a positive greediness, which is the case for $\gamma < 2/3$. However, also Q1 has implications for the choice of γ: As Table VIII shows, for A12 the GM results in a greediness of 0.125, 0.125, and 0.5, and for $\gamma = 0.25$ in 0.375, 0.375, and 0 Therefore, for $\gamma = 0.25$ both allocations allocations have a greediness range of 0.375, which indicates the same fairness for both allocations (the smaller the greediness range the fairer the allocation). However, in the questionnaire A13 was selected more than twice as often as most fair allocation (cf. Section IV-A) compared to A12. Therefore, γ must be chosen, such that the GM indicates a higher fairness for A13 compared to A12, which is the case for $\gamma > 0.25$. Therefore, the questionnaire results give the constraint $\gamma \in]\frac{1}{4}, \frac{2}{3}[$. Within this range, $\gamma = 0.5$ is an appropriate choice, because of the following argument: A13 can be regarded as twice as fair as A12, because A13 got selected by approximately twice as many participants as fairest allocation (cf. Section IV-A). Therefore, γ must be selected such that the GM identifies A13 as twice as fair as A12, *i.e.*, such that the greediness range of A13 is half the greediness range of A12. As Table VIII shows, this is the case for $\gamma = 0.5$.

When the GM is applied subsequently, the used γ value is denoted as superscript behind "GM". For example, when the GM with $\gamma = 0.5$ is applied, it is denoted as applying $\text{GM}_{0.5}$.

C. Resource Normalization

Because different resources occur in different amounts, it is necessary to normalize the offsets before summation (cf. Equation 1). To do so, a normalization constant $k > 0$ needs to be fixed, such that $k/\overleftrightarrow{r_i}$ can be used as normalization factor for every $r_i \in R$. This section discusses two scenarios to motivate the choice of k.

1) Addition of consumers: Let A be an allocation of m resources to the consumers $c_1, \ldots, c_n$ and let $es \in \mathbb{R}^m_{>0}$ be the equal share. Add x times es resources and add x consumers of which each precisely consumes es resources. By this addition, neither the equal share changes nor changes the amount of available resources, *i.e.*, the addition is neutral for the existing allocation. Therefore, the greediness of $c_1, \ldots, c_n$ should not change. To ensure this, k must increase proportionally to the number of consumers in the system, *e.g.*, $k = n$ before the x consumers are added and $k = n + x$ thereafter.

2) Addition of resources: Let $m = 1$, *i.e.*, there is only resource r_1 of which a certain amount is available. Let A^1 be an allocation of r_1 to n consumers. Add y resources and allocate each of these in the same ratio as A^1 allocates r_1. Call this allocation A^y. Then, proportionally A^1 and A^y allocate all consumers the same share of overall resources. Therefore, the greediness of consumers should be the same for both allocations. To ensure this, k must decrease proportionally with the number of resources, *e.g.*, $k = 1$ before the y resources are added and $k = \frac{1}{y+1}$ after the y resources are added.

3) Implication: The considerations above are implemented in the GM by choosing $k = n/m$, where n is the number of consumers and m is the number of resources. This normalization constant could be multiplied by any constant greater zero to scale the resulting greediness. However, multiplying by such constant does not offer additional advantages.

D. Application to Clouds

1) Different Endowments: Above the GM was presented under the assumption that every consumer has the same endowment to the overall resources, wherefore the equal share was used to calculate how many resources consumers cede or over-consume. This section discusses the GM's generalization to individual endowments. To introduce individual endowments, $\overleftrightarrow{r_i}/n$ is replaced by an individual endowment. The sum of endowments must not exceed the overall supply of resources but may be smaller. If the latter is the case, $\frac{\alpha(r_i)}{\beta(r_i)}$ in Equation 2 is replaced by $\min\left(1, \frac{\alpha(r_i)}{\beta(r_i)}\right)$, to avoid this credit factor becomes greater than 1.

This generalization to individual endowments is necessary for the application of the GM to clouds, because VMs have different configurations which must translate to different endowments. The individual endowment of a VM must, therefore, either be defined solely based on the VM's flavor or also based on the resources of the VM's host. In particular, if the latter approach is taken, the endowment can be calculated by partitioning the host resources to the hosted VMs, where VMs receive resources in proportion to their virtual/configured resources. For example, assume a host with 3 GHz and 6GB of hosts two VM 1 and VM 2. VM 1 has 2 VCPU and 4 GB of virtual RAM and VM 2 has 1 VCPU and also 4 GB of virtual RAM. Then the endowment of VM 1 is 2 GHz and 3 GB of RAM and VM 2's endowment is 1GHz and 3GB of RAM.

2) Cloud-wide application: This section discusses how the GM is deployed to enforce fairness between users sharing a cloud. This deployment consists of the following steps that are repeated constantly.

1) Each host calculates the greediness of the VMs that are hosted.
2) The results are announced to the controller node or to all other hosts in a decentralized manner.
3) For each customer the greediness of his VMs is aggregated.
4) Each host regulates the resource access of all hosted VMs, based on the greediness of the respective VM and the greediness of the user owning the VM.

Different functions to implement the aggregation in Step 3 may be chosen. The function used in Section VI is the sum function, *i.e.*, the greediness of a user is the sum of

greediness of his VMs. In Step 4, VMs are assigned priorities to regulate their access to host resources. The priority of a VM is based on the greediness of the customer, who owns the VM. Furthermore, while CPU, Disk I/O and network access are time-shared and can, therefore, be allocated by priorities, RAM is space-shared and can therefore only be allocated by assigning limits. Because it is less flexible to allocate a resource by limits than by priorities, the best suited allocation of RAM needs to be investigated.

3) Incentive Advantages of the VM: This section discusses advantages of the GM compared to the other metrics discussed in Section IV-C1, when applied to ensure fairness in clouds. Unlike these, the GM gives incentive to customers to estimate the resource requirements of their VMs as precise as possible. In particular, the GM gives incentive to chose the configuration of a VM, such that it matches the VMs subsequent load: When a VM tries to exceed the resources it is configured with, it will receive them, if available. However, the customer's greediness will increase with a potentially negative effect on other VMs of the customer. Because a customer's greediness only decreases, when the resources that his VMs did not utilize are utilized by other VMs, it is not guaranteed that unitized resources are credited to the consumer. Therefore, under-, as well as, over-provisioning VMs is costly. No other metric, known so far by today, provides this incentive: when the Price or Price×Scarcity metric is deployed to quantify VM consumption, the cost of customers increases as their VMs utilize resources unrelated to the configuration of the VMs. Therefore, customers have no incentive to fine-tune the configuration of their VMs. When the resource utilization of VMs is unrelated to their configuration, this decreases the cloud's load balancing. In particular, VMs are scheduled based on their configuration, *i.e.*, more "small" than "big" VMs will be scheduled to a host. However, when small VMs face high load, because their owners have no incentive to choose an appropriate configuration, many hosts will become congested. As shown for Scenarios S32 and S33, the price∩scarcity metric even provides an incentive to overspend resources. With the DRF metric, a VM's resource consumption can be related to its configuration. However, because DRF only considers the dominant resource, customers have no incentive to handle other resources economically.

Therefore, the newly developed GM does not only ensure the fair sharing of a cloud, but also gives incentive to customers to wisely configure their VMs. Such an approach allows for the deployment of the cloud efficiently.

VI. Evaluation

This section investigates how the fairness in clouds increases, when resources are allocated according to the GM policy instead of the standard policy. Many existing simulator tools model several resources in the VM scheduling process, especially, the decision, which VM will be started on which host, is made depending on several resources. However, upon simulating the runtime of VMs, only the CPU time is considered. Thus, basic dependencies between CPU time, RAM, disk I/O, and network access are not simulated in existing simulator tools. Therefore, for the initial investigation of the GM, a basic simulator was implemented in Python and is described next.

A. Simulator Design

A simulation is defined by a *setup* that consists of VMs $V = \{v_1, v_2, \ldots, v_p\}$, customers $C = \{c_1, c_2, \ldots, c_n\}$, and hosts $H = \{h_1, h_2, \ldots, h_q\}$. Every VM in V is owned by one customer in C and hosted by one host in H. No customer/host in C/H owns/hosts VMs that are not in V. The VMs compete for m resources on their host. For a VM $v \in V$, $\mathcal{E}(v) \in \mathbb{R}^m_{\geq 0}$ denotes v's endowment (relevant for the GM policy), $\mathcal{D}(v) \in \mathbb{R}^m_{\geq 0}$ denotes v's demand (the demand of each VM is constant over the course of a simulation), $\mathcal{H}(v) \in H$ denotes v's host, and $\mathcal{C}(v) \in C$ denotes v's owner. For a customer $c \in C$, $\mathcal{V}(c) \subset V$ denotes the set of VMs owned by c and $\mathcal{Q}(c) \in \mathbb{R}^m_{>0}$ denotes the quota of c. The endowments c's VMs do not exceed c's quota, *i.e.*, $\forall c \in C\colon \sum_{v \in \mathcal{V}(c)} \mathcal{E}(v) \leq \mathcal{Q}(c)$. To ease the comparison of fairness all customers of a setup have the same quota. For a host $h \in H$, $\mathcal{A}(s) \in \mathbb{R}^m_{>0}$ denotes how many resources are available on h and $\mathcal{V}(h) \subseteq V$ denotes the set of VMs scheduled on h. The endowments of VMs hosted by h do not exceed the resources available on h, *i.e.*, $\forall h \in H\colon \sum_{v \in v(h)} \mathcal{E}(v) \leq \mathcal{A}(h)$.

In the discussed simulations, every host hosts at most one VM of every customer, wherefore a VM is uniquely identified by its owner and host. This allows to increase readability by denoting VM $v \in V$ with $\mathcal{C}(v) = c_b$ and $\mathcal{H}(v') = h_d$ by v^b_d.

The simulator allocates resources according to Leontief utility functions, *i.e.*, VMs receive resources in the ratio of their demand vector. The extension of the experiments to more complex utility functions is future work. Throughout the simulations, the global supply, which is the sum of the supplies of all hosts ($\sum_{h \in H} \mathcal{A}(h)$), is used to normalize resources.

B. Allocation Policies

This discusses the allocation policies that are applied to a setup to generate an allocation.

1) Standard policy: VMs run as processes on their host. Therefore, VMs sharing a host are allocated resources according to the kernel policy. The following mechanism outlines a straight-forward policy to allocate resources to processes with Leontief utility functions: The available resources are allocated equally and each consumer utilizes the maximum amount of this equal share. Because Leontief utility functions are assumed, one resource will limit the utilization of the other resources for each consumer. The resources that are not utilized are pooled and allocated in the same manner. This procedure repeats until $(100 - \epsilon)\%$ of one resource is utilized (in some cases the procedure would have to run indefinitely to allocate all resources). This mechanism provides sharing incentive, is strategy-proof and ensures envy-freeness. Because $(100 - \epsilon)\%$ of one resource may remain unallocated, the mechanism is not Pareto-efficient.Pareto-efficiency can be added by allocating the remaining $(100 - \epsilon)\%$ of the limiting resource (and an proportional amount of the other resources) to an arbitrary customer that demands it. However, then the envy-freeness property is lost.

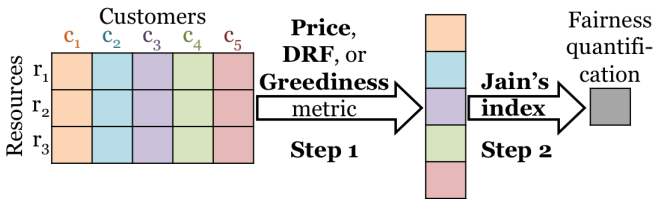


Fig. 2: To quantify the fairness of multi-resource allocations, in Step 1 the bundle of each consumer is mapped to a scalar with the Price, DRF, or Greediness metric, and in Step 2 the fairness of the resulting vector is quantified by Jain's index.

The standard policy is defined by applying this mechanism on every host to allocate the available resources to the hosted VMs.

2) GM$_x$ policy: For $x \in [0,1]$, the GM$_x$ policy allocates resources to the VMs, such that (i) VMs receive resources in the ratio of their demand vector, (ii) no VM receives more resources than requested, and (iii) the minimal greediness of customers is maximized. The greediness of a customer is defined as the sum of greediness of his VMs (cf. Section V-D2). The greediness of VMs is defined by applying the GM$_x$ per host while using the global supply to normalize resources. The Algorithm to find such allocation will be discussed in a future paper.

C. Comparison Method

Because there is no best practice to quantify the fairness of multi-resource allocations, the following two-step approach, which is illustrated in Figure 2, is taken.

1) Step 1: In the Step 1, the resources each customer receives are mapped to a scalar by the following three metrics:

Asset metric The resources received by the VMs of a customer are added up (resulting in a vector) and normalized by the global supply (resulting in a vector). The result is the sum of the elements of the second vector.

DRF metric The resources received by the VMs of a consumer are added up (resulting in a vector) and normalized by the global supply (resulting in a vector). The largest element in this vector is the result.

GM$_x$ metric The greediness of each VM is determined per host according to GM$_x$, while using the global supply to normalize resources. For each customer, the greediness of his VMs is added up and the quota the customer does not utilize is subtracted (normalized by the global supply). Even without the subtraction of the unused quota, this may result in a negative number. Because only nonnegative numbers are suitable in the Step 2, the sum of the initial quota normalized by the global supply vector is added to the result, ensuring nonnegative numbers.

2) Step 2: Each of these metrics in Step 1 results in a vector with a nonnegative entry for each consumer, which quantifies the received resources. Because the quantification of single-resource allocations is well researched [18], [14], the fairness of the vectors resulting from Step 1 is quantified by Jain's index [14] in Step 2. However, other functions would be also feasible (cf. [18]). Therefore, for each metric in Step 1, Step 2 results in a number $q \in [0,1]$ that quantifies the fairness.

D. Notations

When GMs with different γ parameters are enumerated, "GM$_{x,y,z}$" is written instead of GM$_x$, GM$_y$, GM$_z$. Let s be a setup. An allocation a generated by the standard (respectively, GM$_x$) policy is referred to as the "standard (respectively, GM$_x$) allocation for s". Let q be the number that results from the comparison method described in Section VI-C, when metric $M \in \{\text{GM}_x, \text{asset, DRF}\}$ is applied in Step 1. Then, the M fairness of a is q. Furthermore, when a is the standard (respectively, GM$_x$) allocation, then "the standard (respectively, GM$_x$) policy has M fairness q (for s)".

E. Simulation 1: Exchange of Resources on the Same Host

In this setup (cf. Figure 3) there are two resources r_1 and r_2, one host h_a, with $\mathcal{A}(h_a) = (30,30)$, and three customers c_d, c_e, c_f, which all have quota $(20,20)$ and own VM v_a^d, v_a^e, v_a^f, respectively. All VMs are scheduled on h_a and have an endowment of $(10,10)$. The VMs' demands are $\mathcal{D}(v_a^d) = (30,0)$, $\mathcal{D}(v_a^e) = (0,30)$, and $\mathcal{D}(v_a^f) = (30,30)$, *i.e.*, the VM of customer c_d attempts to fully utilize r_1, the VM of customer c_e attempts to fully utilize r_2, and the VM of c_f attempts to fully utilize both resources.

Figure 4 illustrates the standard, and GM$_{0,0.5,1}$ allocations for the setup. The standard allocation would also result from allocating the resources according to DRF. The figure shows, that the GM policy increases the bundle of v_a^f at the cost of v_a^d and v_a^e for smaller γ. This happens because v_a^d and v_a^e have highly asymmetric and complementary demands but symmetric endowments. In particular, both cede one resource entirely, while trying to over consume the other resource. GM$_1$ gives full credit for ceding a resource, allowing for a perfect swap of resources between v_a^d and v_a^e. However, for $\gamma < 1$, the GM does not allow perfect swapping, because ceding is not rewarded as strong as over-consuming is punished. Therefore, the smaller γ, the more resource-swapping increases greediness. Because the GM policy maximizes the minimal greediness in the system, for every swap between v_a^d and v_a^e, v_a^f has to receive a certain amount of resources, such that v_a^f's greediness increases equally. However, the GM$_0$ allocation shows that, even when ceding resources is not credited at all, the resulting GM allocation is still more "balanced" then the standard/DRF allocation.

F. Simulation 2: Cross Host Alignment

For Simulation 2 the setup of Simulation 1 is extended by adding one host that hosts two additional VMs (cf. Figure 3). The additional host, h_b, has the same capacities as h_a. The additional VMs, v_b^d and v_b^f, are owned by customer c_d and c_f, respectively. v_b^d and v_b^f have an endowment of $(10,10)$ and attempt to fully utilize the host, *i.e.*, $\mathcal{D}(v_b^d) = \mathcal{D}(v_b^f) = (30,30)$.

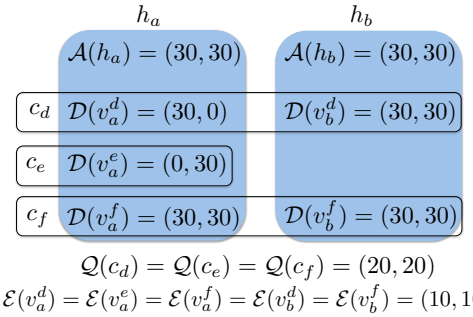
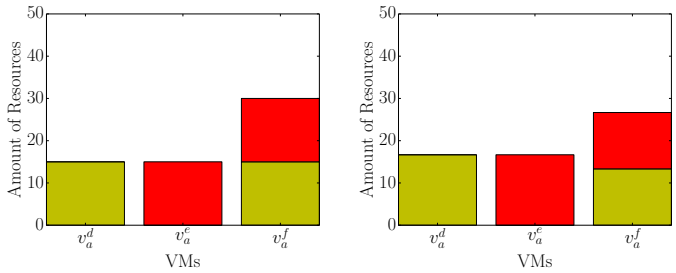


Fig. 3: Illustration of the setups of Simulation 1 and 2. In the setup of Simulation 1 only Host h_a is present, in the setup of Simulation 2 also Host h_b is present.

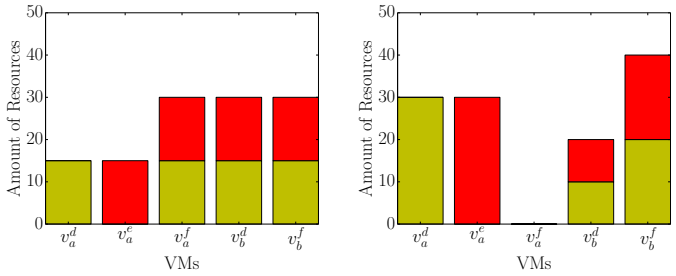


(a) Standard Allocation. (b) GM_0 Allocation.

(c) $GM_{0.5}$ Allocation. (d) GM_1 Allocation.

Fig. 4: Illustration of the standard and $GM_{0,0.5,1}$ allocations for the setup of Simulation 1. The two resources are distinguished by the two colors.

Figure 5 illustrates the standard, and $GM_{0,0.5,1}$ allocations for the setup. As the figure shows, v_a^f is starved by all GM policies for the following reason. Customer c_e is the only customer that only runs one VM, wherefore c_e does not use the entire quota to instantiate VMs. This unused quota is deducted from c_e's greediness. Because, the GM policies maximize the minimum greediness, it is critical to increase c_e's greediness. This can only be done by allocating as many resources as possible to c_e's only VM. However, this implies starving c_f's VM on h_a, and compensating c_f via h_b. While the allocation on h_a is the same for all GMs, the allocation on h_b depends on γ. This is because c_f is forced to cede all resources on



(a) Standard Allocation. (b) GM_0 Allocation.

(c) $GM_{0.5}$ Allocation. (d) GM_1 Allocation.

Fig. 5: Illustration of the standard and $GM_{0,0.5,1}$ allocations for the setup of Simulation 2. The two resources are distinguished by the two colors.

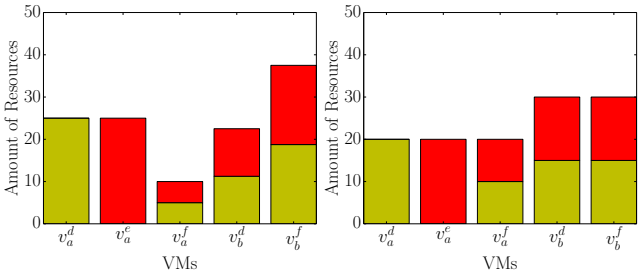
h_a, but the amount with which the ceding is credited changes depending on γ. In particular, GM_1 gives c_f full credit for the forced ceding on h_a and accordingly c_f's share of h_b is larger than for smaller γ. Even though for G_0 c_f does not receive credit for ceding on h_a, c_f's share of h_b is greater than for the standard allocation. This is because because c_d exceeds its endowment on h_a and therefore gets less on h_b.

Because it is neither realistic nor desirable to completely starve VMs, a starvation factor s is introduced. The starvation factor defines the fraction of a VM's endowment that the VM is guaranteed to receive (given it demands the resources). Starvation factors that depend on the customers greediness were also investigated, *i.e.*, the greedier the customer, the smaller the starvation factor for his VMs becomes. However, for these "dynamic" starvation factors dependencies get more complex and are therefore discussed in a future paper. Figure 6 shows the $GM_{0,0.5,1}$ allocations for starvation factors 1 and 0.5.

The allocations for $s = 0.5$ are the same for all GMs for the following reason: The GMs starve v_a^f as much as possible, wherefore, for $s = 0.5$, v_a^f will receive half of its endowment of both resources. Accordingly, v_a^f cedes the other half of both resources, *i.e.*, half of its endowment. Because v_a^d cedes one resource entirely, also v_a^d cedes half of its endowment. Therefore, for $s = 0.5$ the greediness difference of v_a^f and v_a^d is independent of γ. Because greediness difference of v_a^f and v_a^d on h_a is constant, also the resources allocated to the VMs of their owners on h_b is.

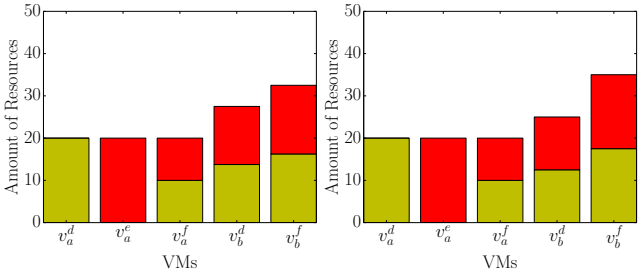
TABLE IX: Fairness comparison of the allocations shown in Figure 4, whereat the fairness of the $GM_{0,0.5,1}$ allocations is given relatively to the standard fairness.

Fairness	Allocation			
	Standard	GM_1	$GM_{0.5}$	GM_0
GM_1	0.889	+0.111	+0.097	+0.058
$GM_{0.5}$	0.980	+0.015	+0.020	+0.015
GM_0	0.997	−0.008	+0.000	+0.003
Asset	0.889	+0.111	+0.097	+0.058
DRF	1.000	−0.074	−0.037	−0.01



(a) $GM_{0,0.5,1}$ Allocations with $s = 0.5$.

(b) GM_1 Allocation with $s = 1$.



(c) $GM_{0.5}$ Allocation with $s = 1$.

(d) GM_0 Allocation with $s = 1$.

Fig. 6: Illustration of the $GM_{0,0.5,1}$ allocations for different starvation factors. The two resources are distinguished by the two colors.

G. Discussion

Table IX compares the asset, DRF, and $GM_{0,0.5,1}$ fairness of Simulation 1. A body cell shows the M fairness of the P allocation, where M and P are given by leftmost column and top row, respectively. The fairness of the $GM_{0,0.5,1}$ allocations is stated relatively to the fairness of the standard allocation. For example, the first body row states, that the GM_1 fairness of the standard allocation is 0.889 and the GM_1 fairness of the $GM_{0,0.5,1}$ allocations are $0.889 + 0.111$, $0.889 + 0.105$, and $0.889 + 0.097$, respectively. Table X compares the fairness of the different allocations of Simulation 2 in the same manner.

Unsurprisingly, Table IX shows that the GM_x fairness of the GM_x allocation is perfect, *i.e.*, 1, for $x \in \{0, 0.5, 1\}$. More interestingly, all GM allocations have higher asset fairness than the standard allocation. Because the standard allocation would also result from allocating the resources according to DRF, the standard allocation achieves perfect DRF fairness. Therefore, in terms of DRF fairness, the standard allocation outperforms all GM allocations. However, as Table X shows, in Simulation 2 *all* GM allocations outperform the standard allocation in terms of *any* fairness. This is because the GM metrics, contrary to the standard policy, allocate resources on individual hosts also based on the allocation on other hosts. Table X also shows that with increasing starvation factor s the $GM_{0,0.5,1}$ and asset fairness of the GM allocation decreases. This is because a higher starvation factor leaves the GM policies less flexibility to allocate resources. However, even for the worse case in terms of flexibility left by starvation limits, *i.e.*, $s = 1$, the fairness is increased by 6 to 6.8 %. For the best case, *i.e.*, $s = 0$, the $GM_{0,0.5,1}$ policies are able to increase the GM_1 and asset fairness by 13.9, 14.8, and 14.9%, respectively.

VII. Summary and Future Work

Multi-resource fairness for the runtime allocation of cloud resources was so far not discussed in literature and not defined satisfactorily. Therefore, the new Greediness Metric (GM) was defined based on the results of a questionnaire on the intuitive understanding of fairness among more than 600 participants. This questionnaire also revealed that other approaches to multi-resource fairness in clouds are not always conform with an intuitive understanding of fairness; this includes DRF, which is the most prominent approach. Because the GM is based on arguments of non-technical participants it is intuitively comprehensible. This makes the GM attractive to be marketed to cloud flat rate customers, as a guarantee to receive the fair share of the leased cloud can be granted.

The GM maps the resource utilization of VMs to a number that quantifies the proportionality of the VMs' resource consumption. By summing up these numbers for customers and allocating resources to VMs based on the sums of their owners, cloud-wide fairness between customers can be achieved. **This allows to define fairness without utility functions** It was shown by simulations, that already for small clouds, the GM outperforms the standard allocation policy in terms of fairness.

However, it also revealed that implementation details still require further research.

VIII. Acknowledgements

This work was supported partially by the SmartenIT and the FLAMINGO projects funded by the EU FP7 Program under Contract No. FP7-2012-ICT-317846 and No. FP7-2012-ICT-318488, respectively.

References

[1] Jörn Altmann, Björn Rupp, and Pravin Varaiya. Effects of Pricing on Internet User Behavior. *Netnomics*, 3(1):67–84, June 2001.

[2] Andrea Arcangeli, Izik Eidus, and Chris Wright. Increasing Memory Density by Using KSM. *2009 Linux Symposium*, Vol. 1, pp 19–28, Montreal, QC, Canada, July 2009.

[3] Thomas Bonald and James Roberts. Enhanced Cluster Computing Performance through Proportional Fairness. *Performance Evaluation*, 79:134–145, April 2014.

[4] Steven J. Brams. *Mathematics and Democracy*. Princeton University Press, Princeton, NJ, USA, 2008.

TABLE X: Fairness comparison of the allocations shown in Figure 5 and Figure 6, whereat the fairness of the $GM_{0,0.5,1}$ allocations is given relatively to the standard fairness.

Fairness	Allocation									
		GM_1			$GM_{0.5}$			GM_0		
	Standard	$s=0$	$s=0.5$	$s=1$	$s=0$	$s=0.5$	$s=1$	$s=0$	$s=0.5$	$s=1$
GM_1	0.821	+0.149	+0.114	+0.068	+0.148	+0.114	+0.066	+0.139	+0.114	+0.060
$GM_{0.5}$	0.933	+0.060	+0.052	+0.035	+0.061	+0.052	+0.036	+0.060	+0.052	+0.035
GM_0	0.987	+0.012	+0.013	+0.006	+0.013	+0.013	+0.010	+0.013	+0.013	+0.012
asset	0.821	+0.149	+0.114	+0.068	+0.148	+0.114	+0.066	+0.139	+0.114	+0.060
DRF	0.926	+0.034	+0.036	+0.022	+0.025	+0.036	+0.032	+0.005	+0.036	+0.038

[5] Abhishek Chandra, Micah Adler, Pawan Goyal, and Prashant Shenoy. Surplus Fair Scheduling: A Proportional-share CPU Scheduling Algorithm for Symmetric Multiprocessors. *4th Conference on Symposium on Operating System Design & Implementation*, OSDI '00, pp 4–4, San Diego, CA, USA, October 2000.

[6] Richard Cole, Vasilis Gkatzelis, and Gagan Goel. Mechanism Design for Fair Division: Allocating Divisible Items Without Payments. *14th ACM Conference on Electronic Commerce*, EC '13, pp 251–268, Philadelphia, PA, USA, June 2013.

[7] Danny Dolev, Dror G. Feitelson, Joseph Y. Halpern, Raz Kupferman, and Nathan Linial. No Justified Complaints: On Fair Sharing of Multiple Resources. *3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pp 68–75, Cambridge, MA, USA, January 2012.

[8] Yoav Etsion, Tal Ben-Nun, and Dror G. Feitelson. A Global Scheduling Framework for Virtualization Environments. *2009 IEEE International Symposium on Parallel Distributed Processing*, IPDPS 2009, pp 1–8, Rome, Italy, May 2009.

[9] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant Resource Fairness: Fair Allocation of Heterogeneous Resources in Datacenters. Technical Report UCB/EECS-2010-55, EECS Department, University of California, Berkeley, CA, USA, May 2010.

[10] Ali Ghodsi, Matei Zaharia, Benjamin Hindman, Andy Konwinski, Scott Shenker, and Ion Stoica. Dominant Resource Fairness: Fair Allocation of Multiple Resource Types. *8th USENIX Conference on Networked Systems Design and Implementation*, NSDI '11, pp 323–336, Boston, MA, USA, March 2011.

[11] Fei Guo. Understanding Memory Resource Management in VMware vSphere 5.0. Performance study, VMware, http://www.vmware.com/files/pdf/mem_mgmt_perf_vsphere5.pdf, Palo Alto, CA, USA, 2011.

[12] Avital Gutman and Noam Nisan. Fair Allocation without Trade. *11th International Conference on Autonomous Agents and Multiagent Systems*, Vol. 2 of *AAMAS '12*, pp 719–728, Valencia, Spain, June 2012.

[13] IBM. Best Practices for KVM. Technical report, http://www.tdeig.ch/linux/pasche/12_BestPractices_IBM.pdf, Austin, TX, USA, November 2010.

[14] Rajendra K. Jain, Dah-Ming W. Chiu, and William R. Hawe. A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems. Technical Report TR-301, Digital Equipment Corp, Hudson, MA, USA, September 1984.

[15] Carlee Joe-Wong, Soumya Sen, Tian Lan, and Mung Chiang. Multi-resource Allocation: Fairness-efficiency Tradeoffs in a Unifying Framework. *31st Annual IEEE International Conference on Computer Communications*, INFOCOM 2012, pp 1206–1214, Orlando, FL, USA, March 2012.

[16] Dalibor Klusáček, Hana Rudová, and Michal Jaroš. Multi Resource Fairness: Problems and Challenges. In: Narayan Desai and Walfredo Cirne (Editors), *Job Scheduling Strategies for Parallel Processing*, Vol. 8429 of *Lecture Notes in Computer Science*, pp 81–95. Springer, Berlin/Heidelberg, Germany, 2014.

[17] Anja Lambrecht and Bernd Skiera. Paying Too Much and Being Happy about It: Existence, Causes, and Consequences of Tariff-Choice Biases. *Journal of Marketing Research*, 43(2):pp. 212–223, May 2006.

[18] Tian Lan, David Kao, Mung Chiang, and Ashutosh Sabharwal. An Axiomatic Theory of Fairness in Network Resource Allocation. *29th Annual IEEE International Conference on Computer Communications*, INFOCOM 2010, pp 1–9, San Diego, CA, USA, March 2010.

[19] Gunho Lee, Byung-Gon Chun, and Randy H. Katz. Heterogeneity-aware Resource Allocation and Scheduling in the Cloud. *3rd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud 2011, pp 4–4, Portland, OR, USA, June 2011.

[20] David Levinson and Andrew Odlyzko. Too Expensive to Meter: the Influence of Transaction Costs in Transportation and Communication. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1872):2033–2046, June 2008.

[21] Andrew Odlyzko. The History of Communications and its Implications for the Internet. *AT&T Labs - Research*, 2000. Odlyzko, Andrew, The History of Communications and its Implications for the Internet. Available at SSRN: http://ssrn.com/abstract=235284 or http://dx.doi.org/10.2139/ssrn.235284.

[22] Andrew Odlyzko. Internet Pricing and the History of Communications. *Computer Networks*, 36(5-6):493–517, June 2001.

[23] David C. Parkes, Ariel D. Procaccia, and Nisarg Shah. Beyond Dominant Resource Fairness: Extensions, Limitations, and Indivisibilities. *13th ACM Conference on Electronic Commerce*, EC 2012, pp 808–825, Valencia, Spain, June 2012.

[24] Sally Floyd, Ed. Metrics for the Evaluation of Congestion Control Mechanisms. RFC 5166, IETF, Berkeley, CA, USA, March 2008.

[25] G. Wei, A. Vasilakos, Y. Zheng, and N. Xiong. A Game-theoretic Method of Fair Resource Allocation for Cloud Computing Services. *The Journal of Supercomputing*, 54(2):252–269, November 2010.

[26] Seyed Majid Zahedi and Benjamin C. Lee. REF: Resource Elasticity Fairness with Sharing Incentives for Multiprocessors. *19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS 2014, pp 145–160, Salt Lake City, UT, USA, March 2014.

[27] Yoel Zeldes and Dror G. Feitelson. On-line Fair Allocations Based on Bottlenecks and Global Priorities. *4th ACM/SPEC International Conference on Performance Engineering*, ICPE 2013, pp 229–240, Prague, Czech Republic, April 2013.

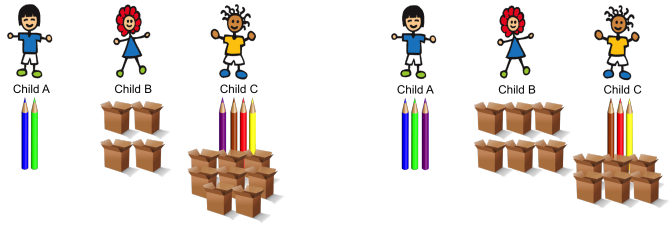
[28] Moshe Zukerman, Liansheng Tan, Hanwu Wang, and Iradj Ouveysi. Efficiency-fairness Tradeoff in Telecommunications Networks. *IEEE Communications Letters*, 9(7):643–645, July 2005.

APPENDIX

A. Choosing the Most Fair Allocation (Q1)

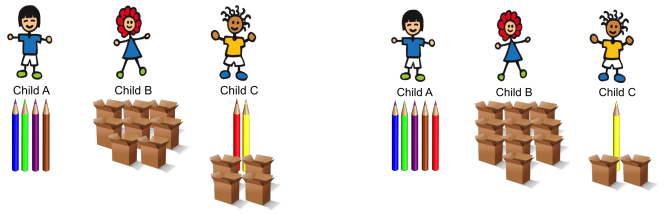
Figure 7 shows the illustrations of A11, A12, A13, and A14 displayed to participants in Q1 of the questionnaire. The question was phrased as follows:

Three children have six colored pencils and twelve cardboard boxes to play with. Because they do not get along well, their nursery teachers decide that they should play on their own, wherefore the painting materials have to be divided



(a) A11

(b) A12



(c) A13

(d) A14

Fig. 7: The illustrations presented in Q1 of the questionnaire.

Profession	Account Manager	Programmer	Administrator
Available	8	8	8
Subsidiary A	2	5	5
Subsidiary B	6	4	4

Fig. 8: The table presented in Q2 of the questionnaire.

among them while taking into account the three following preferences.

- *Child A wants to color, wherefore he only wants pencils (as many as possible).*
- *Child B wants to build cardboard houses, wherefore she only wants cardboard boxes (as many as possible).*
- *Child C wants to color cardboard boxes, wherefore for each pencil he receives, he wants two cardboard boxes to color (Child C wants as many of these sets as possible).*

The nursery teachers are unsure how to allocate the pencils and cardboard boxes among the children and must decide between the following four options.

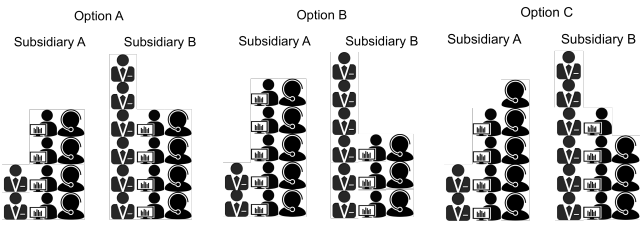
Here the allocations were displayed by the illustrations shown in Table 7 and by tables that depicted the allocations numerically.

Which allocation do you think is the fairest?

B. Allocating based on Requests (Q2)

Figure 9 shows the illustrations of A21, A22, and A23 displayed to participants in Q2 of the questionnaire. The question was phrased as follows:

Human resources of a company are being redistributed. The department heads of two competing subsidiaries A and B must come to an agreement on how 8 account managers, 8 programmers, and 8 administrators are assigned to their departments. The professional groups' workforce does not suffice to meet the department heads' demands simultaneously, which leads to difficult negotiations. After these negotiations, they have almost reached an agreement as depicted in the following table. In particular, the table shows how many professionals the subsidiaries demand after the negotiations. Initially, the demands for accountant managers also exceeded their availability, but this issue was resolved during the negotiations. Here the Table in Figure 8 was shown.



(a) A21 (b) A22 (c) A23

Fig. 9: The illustrations presented in Q2 of the questionnaire.

Since there is still one programmer and one administrator missing to implement the demands, the corporate management has to decide which of the subsidiaries gets one programmer and one administrator less than requested. Since the two subsidiaries are competing, the management wants to find a fair solution.

What do you think is the fairest solution?

- *Subsidiary A gets one programmer and one administrator less. (Option A)*
- *Subsidiary B gets one programmer and one administrator less. (Option B)*
- *One subsidiary gets one programmer less, the other gets one administrator less. (Option C)*

Here the illustrations in Figure 9 were shown.

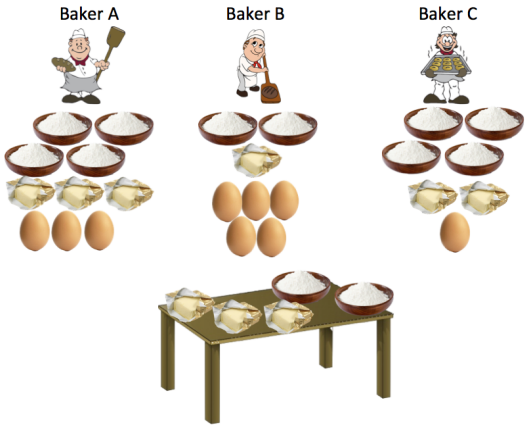
C. Estimating Greediness (Q3)

Figure 10 shows the illustrations of S31, S32, and S33 displayed to participants in Q3 of the questionnaire. The question was phrased as follows:

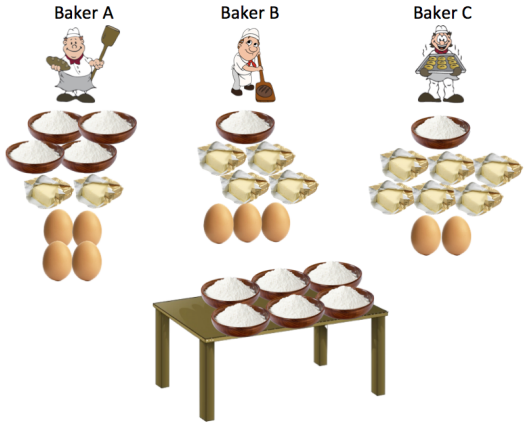
Three bakers each want to bake a cake. To save money, they buy the ingredients (eggs, butter, and flour) together. However, because each baker is using a different recipe, each baker needs a different amount of ingredients.

To split the ingredients, they put the ingredients on a table and each baker helps himself. After a while at least one ingredient is depleted, wherefore the bakers cannot use more of the remaining ingredients; in other words, the ingredients left on the table cannot be utilized due to the lack of depleted ingredients.

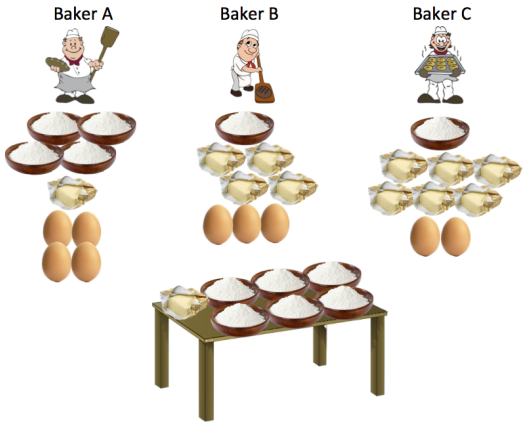
In the following, three scenarios are depicted with what the bakers took from the table and what was available initially. Each scenario is depicted by a table and a graphic, which both demonstrate how much the bakers took of each ingredient.



(a) S31



(b) S32



(c) S33

Fig. 10: The illustrations presented in Q3 of the questionnaire.

Rank the three bakers according to how you perceive their greediness based on the amounts they have used.

Here the allocations were displayed by the illustrations shown in Table 10 and by tables that depicted the allocations numerically.