Financial Evaluation of SLA-based VM Scheduling Strategies for Cloud Federations

Andreas Kohne, Marcel Krüger, Marco Pfahlberg
Materna GmbH
{andreas.kohne, marcel.krueger, marco.pfahlberg}@materna.de

Olaf Spinczyk
Technische Universität Dortmund
olaf.spinczyk@tu-dortmund.de

Lars Nagel
Johannes Gutenberg-Universität Mainz
nagell@uni-mainz.de

Abstract

In recent years, cloud federations have gained popularity. Small as well as big cloud service providers (CSPs) join federations to reduce their costs, and also cloud management software like *OpenStack* offers support for federations. In a federation, individual CSPs cooperate such that they can move load to partner clouds at high peaks and possibly offer a wider range of services to their customers. Research in this area addresses the organization of such federations and strategies that CSPs can apply to increase their profit.

In this paper we present the latest extensions to the *FederatedCloudSim* framework that considerably improve the simulation and evaluation of cloud federations. These simulations include service-level agreements (SLAs), scheduling and brokering strategies on various levels, the use of real-world cloud workload traces and a fine-grained financial evaluation using the new *CloudAccount* module. We use FederatedCloudSim to compare scheduling and brokering strategies on the federation level. Among them are new strategies that conduct auctions or consult a reliance factor to select an appropriate federated partner for running outsourced virtual machines. Our results show that choosing the right strategy has a significant impact on SLA compliance and revenue.

Keywords cloud federation, FederatedCloudSim, simulation, auction, scheduling, SLA, cloud accounting, virtual machine trading

CrossCloud'17, April 23, 2017, Belgrade, Serbia.
Copyright © 2017 ACM 978-1-4503-4934-5...$15.00.
http://dx.doi.org/10.1145/3069383.3069384

1. Introduction

Cloud computing has turned from a hype to an accepted business model in the last years. Besides prevalent models like public, private and hybrid clouds, there is a tendency to utilize or integrate multiple clouds. In *multi-cloud* scenarios, businesses use several clouds, possibly a private cloud within the company and a public cloud in case that the own resources are not sufficient. A very different idea is the *intercloud* [5], the cloud of clouds, in which different cloud service providers (CSPs) use services of one another, especially when their capacity reaches its limit and they need additional resources. When two or more CSPs cooperate in this way and can dynamically use cloud resources from partner clouds, this is called a *cloud federation*. Lately, two large CSPs, *Amazon* and *VMware*, announced a cooperation which allows enterprise customers to migrate their virtual machines (VMs) seamlessly between their data centers [1]. Furthermore, the open-source cloud platform *OpenStack* was extended to support federations of CSPs [10].

The foundation for cloud federations are service-level agreements (SLAs) between the cooperating CSPs. Such SLAs regulate under which conditions services and resources are provided, *i.e.*, they specify the services, costs and penalty payments. It is important that these SLAs are not just written contracts between the CSPs but that they are machine-readable. This allows to fully automate the processes of service brokering, VM migrations and payment transactions. Automation leads to stable service quality and financial benefits. To further increase the profit, CSPs must try to observe all SLAs because broken SLAs entail penalty payments and customers must be refunded. This can be achieved by providing enough resources to the VMs and minimizing the number of migrations in and between DCs and CSPs as they directly impact the uptime and availability of VMs.

As research with real clouds is very complex and expensive, *simulations* are a good way to investigate different cloud scheduling strategies and their impact on SLA compliance and financial aspects. In this paper we present the latest extensions to *FederatedCloudSim* (FCS) and demonstrate how this simulation framework can be used to evaluate scheduling and brokering strategies. Our contributions are:

- We extended FederatedCloudSim by a cloud accounting module (*CloudAccount*) such that one can better analyze the financial impact of scheduling and brokering strategies in cloud federations.
- New scheduling strategies are added to FCS. Now partner clouds for outsourcing VMs can, for example, be determined by an auction or by consulting a reliance factor based on the VM trading history.
- To make FCS simulations more relevant and close to real-world scenarios, we use genuine cloud workload traces from *Bitbrains* and *Materna.*We recently released the Materna traces for scientific purposes.
- We compare several strategy combinations in a scenario with five CSPs by changing only the combination of one CSP. By doing so, we demonstrate what effects the strategies have and which objectives they achieve.

This paper is structured as follows: After discussing related work in Section 2, we describe the extensions of FederatedCloudSim and the scheduling strategies in Section 3 and 4. Finally, in Section 5 we simulate different scenarios and analyze the impact of strategies on SLA compliance and CSP finances, before we conclude the paper in Section 6.

2. Related Work

Scheduling and brokering in clouds and cloud federations have been hot research topics in recent years. There is a multitude of papers about scheduling VMs in a single cloud [3, 19, 21] and about how CSPs can benefit from joining a federation [4, 16]. In the following we concentrate on strategies for the exchange of VMs in federations.

Such strategies must choose VMs to trade and partners to trade with for which they can check criteria like outsourcing prices, SLAs, security, current loads, energy efficiency or energy prices. There are more static strategies that assess mentioned criteria applying rule-based approaches, game theory or queuing theory [5, 7, 9, 12] and more dynamic ones which include negotiations / auctions [2, 8, 11, 17, 18, 20].

Most of the work simplifies the scenarios by having only one basic scheme for internal scheduling (like *round robin*), by allowing no more than one data center per CSP or by ignoring SLAs. Instead of using cloud traces, the simulations are often run on grid traces consisting of batch jobs. Our framework *FederatedCloudSim* allows cloud trace-driven simulations of SLA-based federated clouds with multiple data centers and schedulers on every level. We deem it important to combine strategies and test if they go well together.

Survey of Scenarios and Strategies *InterCloud*, introduced by Buyya *et al.* [5], is a federation with a central *Cloud Exchange* which functions as a marketplace for VMs. Every provider communicates with the Exchange via a *Cloud Broker* and has a local *Cloud Coordinator* that schedules, but does not migrate VMs inside the local data center. Their CloudSim simulations show that federations considerably reduce waiting and computing times. Another approach by Celesti *et al.* [7] uses *Cross-Cloud Federation Managers* which perform three tasks: (1) *Discover*: Search for federated partners via peer-to-peer approach. (2) *Match Making*: Selection of best partner based on predefined criteria. (3) *Authentication*: *Single sign-on* authentication of customer which is then provided VMs by the partner selected. The VMs do not come with SLAs and are not monitored. Decisions are made by a central instance based on fixed criteria.

Darzanos *et al.* [9] try a more theoretical approach by applying queuing theory. Each CSP, having its own client, is modelled as an M/M/1 queue and follows a possibly selfish policy respecting energy prices and quality of service requirements. A policy is a tuple $(\alpha_1, \alpha_2, ..., \alpha_n)$ which specifies what portions are sent to the other CSPs. $(0.8, 0.2)$ would, for example, mean that CSP 1 keeps 80% of the jobs and gives the rest to CSP 2. The aim is to find policies for all CSPs so that an objective function is optimized. The authors limit their investigation to the case $n = 2$.

Chang *et al.* [8] present a simple auction-based algorithm respecting SLAs for job resource provisioning in multi-cloud environments. They assume job deadlines, use a cost model similar to ours and evaluate the jobs' makespan and price. Evaluating their algorithm on a small *OpenStack* environment, they show an increase in revenue. Li *et al.* [17] designed a double-auction-based mechanism for exchanging VMs between cloud providers. In every auction the CSPs act as sellers and buyers stating the prices they demand or are willing to pay for VMs of certain types. A central broker functioning as an auctioneer determines at which prices the VMs are sold. The bids are computed by a dynamic trading and scheduling algorithm that takes SLAs and energy prices into consideration. Majhi and Bera [18] suggest Dutch and English auction methods with multiple rounds in which the bidders and sellers can update their prices / bids. The auctions respect

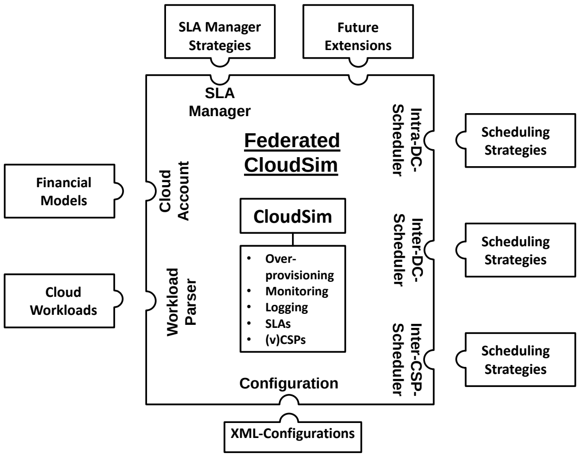

Figure 1. Structure of FederatedCloudSim

business agreements between the CSPs, but there are no SLAs between user and CSP. Tang and Chen [20] consider a broker-based market where cloud providers can offer VMs to others. Their scheme uses an iterative double-auction mechanism and incorporates properties of federations like the average response time. Fard *et al.* [11] introduce a pricing model and a truthful scheduling mechanism for grid / cloud computing considering two objectives: monetary cost and completion time. Assuming selfish providers who may give false information about their resources, the authors compare their new reversed auction-based scheduler to two multi-objective evolutionary algorithms and show in trace-driven GridSim simulations that their approach is superior. In contrast to this work, they use a simple financial and SLA model and assume that CSPs know about the resources of their partners. Anisetti *et al.* [2] describe e-auctions for multi-cloud service provisioning. They evaluate a service selection process based on these auctions where customers want to find the cheapest and most trustworthy CSP using matching and ranking algorithms. In contrast to our work, CSPs do not trade VMs with one another, but instead the auction is started by the user.

3. Extensions of FCS

Scheduling strategies and other aspects of cloud federations are usually evaluated using cloud simulation software. The software that we use and extend in this paper is *FederatedCloudSim* (FCS) [13–15] which originated from the *EASI-CLOUDS* project [22]. It is based on *CloudSim* [6] and implements a model for multi-level VM schedulers especially designed for cloud federations [13]. Its main features are:

Simulation of federation scenarios: the scenarios can include CSPs with an arbitrary number of data centers and virtual cloud service providers (vCSPs) which act as service brokers without resources of their own.

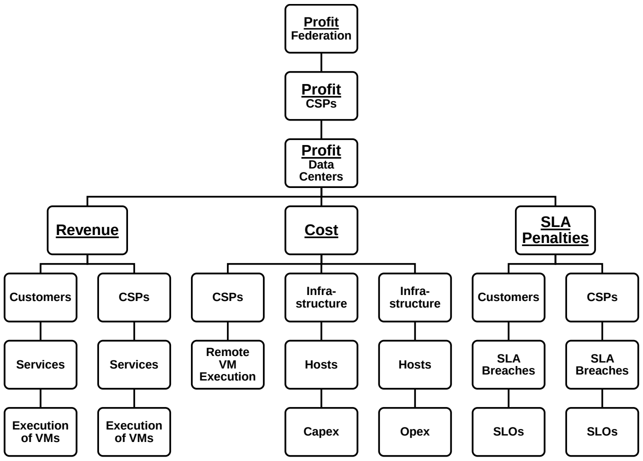

Figure 2. Financial model *CloudAccount*

Service-level agreements: SLAs can be defined between CSPs and between CSP and customer; FCS provides fully automated SLA and service request management.

Three-level VM scheduling: intra-DC and inter-DC level scheduling at a single CSP; inter-CSP level VM brokering in cloud federation.

Pre-built scheduling strategies: there is a selection of algorithms for every level; more strategies can be easily added.

Resources: VMs are assigned RAM and CPU; resource overprovisioning is possible.

Workload traces: FCS allows for processing real-world cloud workload traces.

Monitoring: events and resources can be logged and filtered for evaluation.

Like CloudSim, FCS is modular which makes it flexible and extensible. New strategies and generally new functionality are added by extending abstract classes which provide the core functionality for communicating with the framework. The schematic structure of FCS is shown in Figure 1.

The main new features of the latest version are:

CloudAccount: To evaluate the impact of different scheduling strategies on CSP revenue and profit, a flexible and extensible finance model was added (Figure 2). The current implementation includes the main flows of funds: revenue, costs, SLA penalties. CloudAccount scales to different levels and offers financial insight into DCs, CSPs or the complete federation. We implemented an accounting strategy that is responsible for filling the flows of funds with concrete data. Therefore it evaluates the specific usage of ressources (CPU and RAM) and number of SLA breaches of executed and outsourced VMs as well as the number of hosts to calculate revenue, cost and penalties. In order to value these technical metrics money-wise financial parameters are to be set in the XML configuration file. These parameter reflect the agreements as they are determined for each provider with its customers and federation partners in the respective SLAs.

Auction platform: Using the CloudAccount functionality to model dynamic price changes during simulation runtime, an extensible auction platform was implemented that allows federated CSPs to auction VMs. The platform provides interfaces for auction-based schedulers as well as bidding strategies. By implementing these interfaces, one can add new auction and bidding strategies.

Additional scheduling strategies: Besides auctions, one can use alternative strategies for exchanging VMs on the federation level. Some of these strategies are described in Section 4.

Furthermore, the SLA management was extended to the whole federation and the logging mechanism to include the CloudAccount data.

4. Scheduling Strategies

For each of the three scheduling levels, we implemented several strategies. In this section we shortly describe all strategy combinations used in the evaluation (Section 5).

Migrations are triggered if an SLA is already broken or if an SLA breach is imminent. Every CSP has an SLA manager that monitors load thresholds for every host to detect such events. If more than 80% of a given resource (RAM or CPU) is in use, a host is flagged as *at risk*, and if the utilization does not decrease within the next simulation tick, the local scheduler is called to reschedule some of the host's VMs. Otherwise, *i.e.*, if the load drops below 80%, no action is performed. In case of a sudden rise above 90%, an immediate reschedule is triggered.

On the lowest scheduling level, the inter-DC level, the DC scheduler selects the VMs for migration and, if possible, destination hosts in the same data center. In the simulations we use the following two strategy combinations (in the following usually referred to by their short name):

First Fit + First Fit (FirstFit): According to the host's current list of VMs, the host's first VMs are migrated to other hosts, until the threshold violation is resolved. Each VM is migrated to the first host that has sufficient resources left.

Minimization of Migrations + Worst Fit (MMWF): The minimum amount of VMs necessary to solve the (imminent) violation is selected and one by one migrated to the host with the least utilization (unless all of them are full).

A VM is only migrated within the same DC if there is at least one host that can take it without exceeding its own thresholds. If no such host is found, the *DC broker* is activated which, as a central entity, manages all the DCs of its CSP. Based on its *global* strategy, it can overrule the *local* selection of the DC scheduler and arbitrarily alter the set of VMs to migrate and their destinations. The global strategy should aim at optimizing the VM allocation over all hosts with respect to the CSP's criteria (*e.g.*, energy usage or throughput) and could run contrary to a DC scheduler's local strategy. In the simulations we use the following four strategy combinations:

First Fit + First Fit (FirstFit): This strategy adopts the preselected set of VMs from the DC scheduler and selects the first data center from the list of data centers that has sufficient free resources.

Minimization of Migration + Least Threatened (MMLT): The VMs are selected according to Minimization of Migration (and are equal to the preselection of the DC scheduler if it follows MMWF). As destination the strategy chooses the *least threatened data center* which we define as the one that is least likely to violate the downtime limit given in the SLA.

Lowest SLA Category + Least Threatened (LSLT): This strategy migrates the CSP's VMs with the lowest SLA category to the least threatened data centers. The *SLA categories* are a construct that simplifies the configuration of VMs and their SLAs. When a new VM is created, one can simply assign a predefined or a new category to it which specifies the amount of requested resources, the agreed availability, the limit for the number of migrations and the cost for VM execution.

Least Threatened + Least Threatened (LTLT): The strategy migrates the CSP's VMs first whose downtime is well above the downtime limit defined in the SLA and picks the least threatened data center as destination.

If the DC broker is unable to resolve the problem, the *cloud broker* on the third scheduling level, the inter-CSP level, is triggered. It does not gather any additional information about the data centers and simply keeps the VM selection. The cloud broker acts as a communication interface between the data centers of the same provider and the other providers of the federation. It receives requests to migrate VMs to federated partners and determines an appropriate CSP by evaluating information about the partners.

As it is very likely in real federations that the companies involved do *not* share details about their infrastructure, utilization, scheduling strategies or financial constitution, we assume a decentralized structure where each CSP has a very limited view of his federated partners. The usable information is essentially the financial

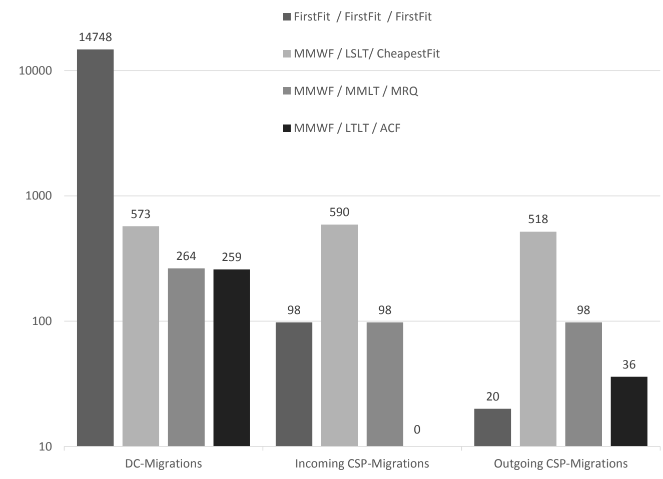

Figure 3. Comparison: Number of migrations of different scheduling strategies (logarithmic scale)

parameters for violations and VM acquisition defined in the SLA. These are, however, also the key figures which influence the profitability of outsourcing VMs and which our strategies exploit. In order to evaluate the amount and type of SLA breaches, we extended the FCS framework by a rating component which a CSP can use to calculate how reliable his partners are. By keeping statistics about the migrated VMs, the rating component can assess the partner based on (weighted) parameters like successfully executed VMs, downtime, SLA breaches due to downtime or SLA breaches due to a lack of resources (CPU or RAM). Additionally, such occurences can also be weighted by their age. In this way, providers can compare federated partners with respect to their reliability to execute outsourced VMs and use this information as a basis for selecting a migration destination. In the simulations we use the following four strategies:

FirstFit: This strategy takes the first CSP from the list of CSPs that agrees to run the VM.

CheapestFit: Based on the SLA parameters for outsourcing costs and penalty fees, the cheapest partner is chosen.

MostReliableQuotation (MRQ): In the first of two phases, a subset of partners is chosen that offer (relatively) low resource prices and high compensations in case of SLA violations. In the second phase, the strategy picks the most reliable CSP from this subset.

AuctionCheapestFit (ACF): The new auction platform is used to conduct a concealed auction with two bidding rounds. Every federated partner receives an invitation to join the auction and to make a bid for an offered VM. After the first round, every attendee is informed whether his bid is the best one. After the second bidding round, the auctioneer picks the cheapest provider which receives the VM.

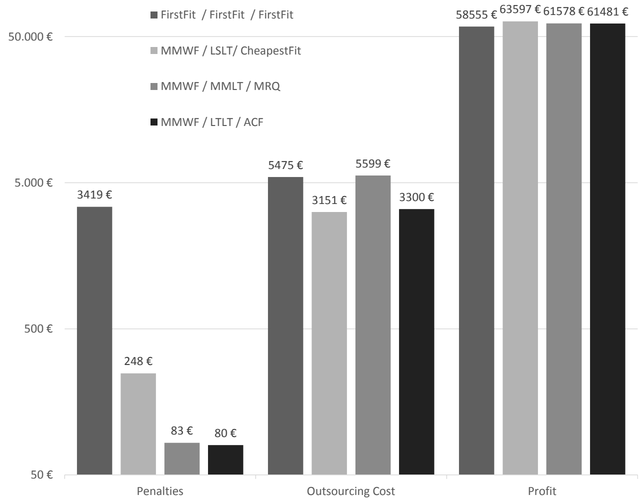

Figure 4. Comparison: Accounting results of different scheduling strategies (logarithmic scale)

FairFit: Like FirstFit, this strategy takes the first CSP from the list of CSPs that agrees to run the VM. However, it does not always start from the beginning of the list like FirstFit, but runs through the list in round-robin fashion.

5. Evaluation

In order to compare the impact the different scheduling strategies have, we ran a set of four simulations with five providers each having one data center. One provider modifies its strategy combination while the other four CSPs keep theirs. For each CSP we use a different configuration and trace which are summarized in Table 1. Due to lack of space, we do not describe the providers' penalties and accounting factors in this paper.

Providers 2, 3, 4 and 5 use the same combination of strategies in all four simulations. Their DC schedulers use MMWF, their DC brokers FirstFit and their cloud brokers FairFit. The first provider, on the other hand, uses a different combination in each simulation, see Table 2. All five providers have a dynamic bidding strategy (used in Sim. 4) that computes the bid value dependent on the provider's resource utilization.

5.1 Results

To compare the four strategy combinations, we ran the simulations outlined above and analyzed the logs of CSP 1 regarding penalty payments, outsourcing costs and profit as well as the number of DC migrations (*i.e.*, at the CSP) and incoming and outgoing CSP migrations (*i.e.*, from / to partners). The migrations are plotted in Figure 3, the financial results in Figure 4.

The combination of FirstFit strategies on all three scheduling levels is used as baseline; it does not aim at improving any metric. The MMWF strategy – used by

Table 1. Configurations of providers

CSP	hosts	$\frac{cores}{host}$	$\frac{GHz}{core}$	$\frac{RAM}{host}$	trace (# VMs)
1	49	30	2.9	128GB	Bitbrains A (1250)
2	36	30	2.9	128GB	Bitbrains B (500)
3	12	30	2.9	100GB	Bitbrains C (500)
4	11	30	2.9	128GB	Materna A (520)
5	20	30	2.9	96GB	Materna B (527)

Table 2. Strategies of providers

CSP	Sim.	DC Scheduler	DC Broker	Cloud Broker
1	1	FirstFit	FirstFit	FirstFit
1	2	MMWF	LSLT	CheapestFit
1	3	MMWF	MMLT	MRQ
1	4	MMWF	LTLT	ACF
2-5	1-4	MMWF	FirstFit	FairFit

the DC scheduler of all other combinations – targets the reduction of migrations between hosts and fulfills this task very convincingly decreasing them from 14748 (for the FirstFit strategies) to at most 573 in any other combination. As we will see, this reduction of at least 96.1% has a major influence on CSP 1's (downtime) penalty payments.

In the second combination of strategies, the downtime penalty costs are further reduced by the DC broker strategy LSLT which prefers migrating VMs of a low SLA category. Such VMs are allowed to be migrated more often than higher classified ones, and the penalties for SLA violations are also lower. The CheapestFit strategy on the inter-CSP level aims at reducing costs by taking the cheapest offer. As a result, the penalties are cut by 92% and the outsourcing costs by 42.5% (compared to the baseline).

The third combination includes the DC broker strategy MMLT which aims at reducing penalties and outsourcing costs by simply decreasing the overall number of migrations. The number of DC migrations is 98.2% lower than in the first and 53.9% lower than in the second simulation. The idea of the cloud broker strategy MostReliableQuotation is to reduce penalties and outsourcing costs by choosing the most reliable CSP from a set of low-cost CSPs. This helps to cut the penalty payments by 97.6% and 66.5% in comparison with Combination 1 and 2, respectively. But outsourcing costs are actually higher because of ignoring SLA categories and not picking the cheapest destination.

In the fourth simulation, the LTLT strategy reduces penalty payments by migrating the least threatened VMs first. Hence, the downtime is better distributed and more balanced over the entirety of VMs which leads to less SLA violations. Compared to the first three simulations, the fourth one achieves a reduction of 97.7%, 67.7% and 3.6%, respectively. Using an auction-based strategy on the inter-CSP level is immediately reflected in the outsourcing costs because it selects the cheapest partner and the dynamically determined charges can be even lower than the ones set in the SLAs. The outsourcing costs are 4.5% higher than in Simulation 2, but 39.7% and 41.1% lower than in Simulation 1 and 3. Another aspect of the dynamic bidding strategies is that the provider with the most free resources tends to offer the best bid. As a consequence, the load is more balanced among the providers and overallocation is rarer which makes migrations and penalties less likely. Adding incoming and outgoing CSP migrations together, the fourth strategy combination cuts the number of migrations by 69,5%, 96.8% and 81.6% compared to Simulation 1, 2 and 3, respectively.

All of the last three strategy combinations used different ways to outperform the simple baseline configuration and succeeded increasing the overall profit by 8.6%, 5.2% and 5.0%, respectively.

6. Conclusion and Outlook

In this paper we have presented the latest extensions to *FederatedCloudSim*, a software framework for simulating cloud federations. The extensions include the financial model *CloudAccount* and a range of strategies for migrating and trading virtual machines within and in between clouds. CloudAccount combines revenues, costs and penalties in a tree-like structure and make it easier to organize and follow the flows of funds. The new strategies include an auction platform that CSPs can use to "sell" their VMs to the highest bidder.

Using FederatedCloudSim we have set up a scenario and compared different federation schedulers with respect to SLA compliance and financial impact. The results show how the different strategy combinations applied influence data center utilization, VM migration and trade and, thus, revenue and penalties. As one can see, choosing an appropriate strategy has a huge impact.

However, this analysis is limited due to the lack of space, for which reason we are planning to extend our analysis in future work. Since the results depend on the scenario, the policies of the partners and the strategy combination of the provider in question, we will take all these factors into consideration to provide a more detailed evaluation.

7. Closing Remarks

This work was supported by Materna GmbH and by the German Ministry of Education and Research under Grant 01IH13004 ("FAST"). Materna is a full-service IT provider in the premium segment and has been successfully implementing ITC projects for their customers for more than 35 years.

The Materna traces we used were recorded in one of their German data centers and consist of three data sets each reflecting one month of data of more than 500 business-critical VMs. The traces have been made public and are available at: http://gwa.ewi.tudelft.nl/datasets/gwa-t-13-materna.

The FederatedCloudSim framework and all schedulers are available at the FCS project site: http://ess.cs.uni-dortmund.de/EN/Research/Projects/FederatedCloudSim/index.html.

References

[1] VMware Cloud on AWS: VMware Will Deliver SDDC Technologies on AWS. `https://www.vmware.com/cloud-services/vmware-cloud-aws.html`, 2017. Accessed: 2017-01-20.

[2] M. Anisetti, C. A. Ardagna, P. A. Bonatti, E. Damiani, M. Faella, C. Galdi, and L. Sauro. e-Auctions for Multi-cloud Service Provisioning. In *International Conference on Services Computing (SCC)*, pages 35–42. IEEE, 2014.

[3] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. *Future Gener. Comput. Syst.*, 28(5):755–768, May 2012. ISSN 0167-739X.

[4] D. Bermbach, T. Kurze, and S. Tai. Cloud federation: Effects of federated compute resources on quality of service and cost. In *International Conference on Cloud Engineering (IC2E)*, pages 31–37. IEEE, 2013.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In *Algorithms and architectures for parallel processing*, pages 13–31. Springer, 2010.

[6] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Software: Practice and Experience*, 41(1):23–50, 2011.

[7] A. Celesti, F. Tusa, M. Villari, and A. Puliafito. How to enhance cloud architectures to enable cross-federation. In *3rd International Conference on Cloud Computing (CLOUD)*, pages 337–345. IEEE, 2010.

[8] C.-C. Chang, K.-C. Lai, and C.-T. Yang. Auction-based Resource Provisioning with SLA Consideration on Multi-cloud Systems. In *Computer Software and Applications Conference Workshops (COMPSACW), 2013 IEEE 37th Annual*, pages 445–450. IEEE, 2013.

[9] G. Darzanos, I. Koutsopoulos, and G. D. Stamoulis. A model for evaluating the economics of cloud federation. In *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pages 291–296, Oct 2015.

[10] J. A. L. del Castillo, K. Mallichan, and Y. Al-Hazmi. Openstack federation in experimentation multi-cloud testbeds. In *5th International Conference on Cloud Computing Technology and Science (CloudCom)*, volume 2, pages 51–56. IEEE, 2013.

[11] H. M. Fard, R. Prodan, and T. Fahringer. A truthful dynamic workflow scheduling mechanism for commercial multicloud environments. *IEEE Transactions on Parallel and Distributed systems*, 24(6):1203–1212, 2013.

[12] F. Jrad, J. Tao, and A. Streit. SLA Based Service Brokering in Intercloud Environments. In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science (CLOSER)*, pages 76–81, 2012.

[13] A. Kohne and O. Spinczyk. Model for SLA-Based VM Scheduling in Federated Cloud Environments. *J. Integrated Design & Process Science*, 18(1):39–52, 2014.

[14] A. Kohne, M. Spohr, L. Nagel, and O. Spinczyk. FederatedCloudSim: A SLA-aware Federated Cloud Simulation Framework. In *Proceedings of the 2nd International Workshop on CrossCloud Systems (CCB@Middleware)*, pages 3:1–3:5, 2014.

[15] A. Kohne, D. Pasternak, L. Nagel, and O. Spinczyk. Evaluation of SLA-based Decision Strategies for VM Scheduling in Cloud Data Centers. In *Proceedings of the 3rd Workshop on CrossCloud Infrastructures & Platforms*, CrossCloud '16, pages 6:1–6:5, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4294-0.

[16] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze. Cloud Federation. *CLOUD COMPUTING*, 2011:32–38, 2011.

[17] H. Li, C. Wu, Z. Li, and F. C. M. Lau. Virtual Machine Trading in a Federation of Clouds: Individual Profit and Social Welfare Maximization. *IEEE/ACM Trans. Netw.*, 24(3):1827–1840, June 2016. ISSN 1063-6692.

[18] S. K. Majhi and P. Bera. VM migration auction: Business oriented federation of cloud providers for scaling of application services. In *2014 International Conference on Parallel, Distributed and Grid Computing*, pages 196–201, Dec 2014.

[19] J. Sekhar, G. Jeba, and S. Durga. A Survey on Energy Efficient Server Consolidation Through VM Live Migration. *International Journal of Advances in Engineering & Technology*, 2012. ISSN 2231-1963.

[20] L. Tang and H. Chen. Double auction mechanism for request outsourcing in cloud federation. In *2015 IEEE International Conference on Communication Workshop (ICCW)*, pages 1889–1894, June 2015.

[21] S. Tilak and D. Patil. A Survey of Various Scheduling Algorithms in Cloud Environment. *International Journal of Engineering Inventions*, 1(2):36–39, 2012. ISSN 2278-7461.

[22] J. Wu, K. Narayanan, L. Nagel, C. Fiehe, A. Litvina, J. Tonn, C. Zoth, H. Goltz, S. Unger, F. Pursche, M. Scheel, A. Brinkmann, and W. Thronicke. Building a Medical Research Cloud in the EASI-CLOUDS Project. *Concurrency and Computation: Practice and Experience*, 27(16):4465–4477, 2015.