

Evaluation of SLA-based Decision Strategies for VM Scheduling in Cloud Data Centers

Andreas Kohne
Damian Pasternak

Materna GmbH
{andreas.kohne,damian.pasternak}
@materna.de

Lars Nagel

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

Olaf Spinczyk

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

Abstract

Service level agreements (SLAs) gain more and more importance in the area of cloud computing. An SLA is a contract between a customer and a cloud service provider (CSP) in which the CSP guarantees functional and non-functional quality of service parameters for cloud services. Since CSPs have to pay for the hardware used as well as penalties for violating SLAs, they are eager to fulfill these agreements while at the same time optimizing the utilization of their resources.

In this paper we examine SLA-aware VM scheduling strategies for cloud data centers. The service level objectives considered are resource usage and availability. The sample resources are CPU and RAM. They can be overprovisioned by the CSPs which is the main leverage to increase their revenue. The availability of a VM is affected by migrating it within and between data centers. To get realistic results, we simulate the effect of the strategies using the FederatedCloudSim framework and real-world workload traces of business-critical VMs.

Our evaluation show that there are considerable differences between the scheduling strategies in terms of SLA violations and the number of migrations. From all strategies considered, the combination of the *Minimization of Migrations* strategy for VM selection and the *Worst Fit* strategy for host selection achieves the best results.

Categories and Subject Descriptors CR-number [*subcategory*]: third-level

General Terms scheduling, SLAs, cloud computing, simulation

Keywords cloud computing, simulation, scheduling, service level agreement

1. Introduction

While there is a lot of research on the enforcement of service level agreements (SLAs) in the context of *grid computing* and *cloud computing* (e.g. [4, 10, 17]), most real-world *cloud service providers* (CSPs) offer only few and lightweight SLAs because it is difficult to measure and enforce them [9]. But SLAs gain importance for commercial CSPs because many customers demand guarantees and the possibility to monitor the service execution and the resources assigned to their virtual machines (VMs).

The EASI-CLOUDS research project¹ considers SLA-aware scheduling and brokering strategies in clouds and cloud federations and the automatic negotiation of SLAs between customers and CSPs [13, 21]. As testing in real world environments is complex and expensive, we have extended the simulation tool CloudSim [6] for our purposes. The first implementation of *FederatedCloudSim* was presented in [14]. This framework has now been extended so that it allows for monitoring and enforcing SLAs regarding VM availability and resources (like CPU and RAM).

In this paper FederatedCloudSim is used to investigate SLA-aware strategies for VM placement in cloud data centers. We consider combinations consisting of a strategy for selecting VMs to migrate and a strategy for selecting the new hosts. Our results compare these strategies with respect to SLA violations and the number of VM migrations. In the simulation we allow migrations of VMs within and in between data centers of the same CSP which are triggered in case of (imminent) SLA violations. To the best of our knowledge, this is the first publication on SLA-aware VM scheduling using real world cloud datacenter traces of business critical machines like database, web and financial application servers.

[Copyright notice will appear here once 'preprint' option is removed.]

¹ <http://easi-clouds.eu/>

In the remainder of this section we give an overview of the related work on scheduling strategies and cloud simulation frameworks. In Section 2 we describe the extensions to the FederatedCloudSim framework used for the experiments. In the experiments we investigate SLAs and scheduling strategies that are explained in Section 3. The experiments are evaluated in Section 4. The paper is concluded in Section 5 by summarizing the results and outlining future work.

1.1 Related Work

In this section we will first describe work related to SLAs and scheduling strategies in the area of cloud computing. In the second part we will shortly describe alternative cloud simulation software.

Strategies and Service Level Agreements in Simulations

Since guaranteed service levels gain importance, cloud SLAs are an active area of research. In [20] Van et al. investigate the SLA-aware management of virtual resources in cloud infrastructures. Comuzzi et al. [7] describe an SLA framework for cloud computing, which was implemented in their research project *SLA@SOI*. They focus on the automation of SLA management and contract making. The research by Jrad et al. is close to our *EASI-CLOUDS* project as they define a scenario in which users ask a cloud broker to find the most appropriate CSP for the service and SLAs required. In [11] they use a *CloudSim* extension their experiments in which they consider SLA negotiation with a centralized broker, but no strategies for scheduling VMs in data centers.

Marer et al. [1] present an SLA manager for CSPs that categorizes VMs using five escalation levels where the level determines the action performed by the VM scheduler. Their simulation results show that their rule-based approach ensures more SLAs compared to an approach using *case-based reasoning* (CBR). In our simulation framework we use a similar approach, but split the decision process into two parts. First the SLA manager checks for every SLA whether it is close to being violated. If so, the local VM scheduler is informed and decides whether to reschedule the respective VM or to migrate it to another data center.

Lu et al. [15] describe and implement SLA-aware VM placement strategies that minimize the number of migrations. In their CloudSim-based simulation they investigate the trade-off between higher service availability and lower energy consumption. While migrations decrease the availability, they can help to consolidate the virtual machines on few servers so that the remaining ones can be run in energy saving mode. They use a basic implementation of CloudSim with a standard resource usage model for simulating VM workloads. In this paper we present simulation data using an SLA manager that tries to maintain all SLAs. Furthermore we use real-world cloud traces to generate more accurate results.

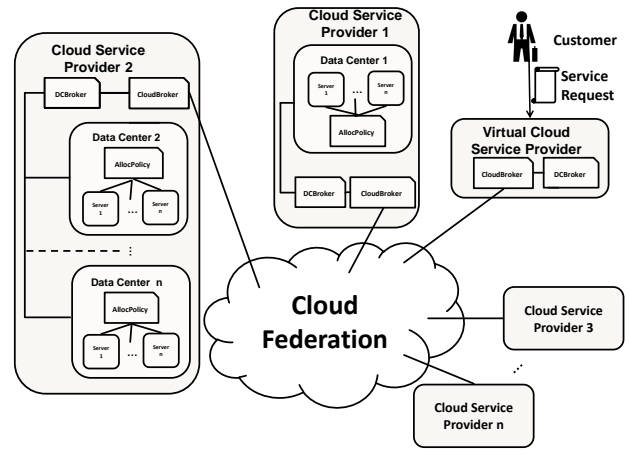


Figure 1. Federated cloud scenario

Beloglazov et al. examine different strategies for scheduling VMs in an energy-efficient way [2, 3]. The strategies detect overloaded hosts and select VMs to migrate. They simulate scenarios using CloudSim and compare the strategies with respect to energy consumption. In their papers they only perform data center scheduling and do not consider SLAs.

In this paper we apply some of their algorithms, but integrate an SLA management and a multi data center environment. We present first results for our multi-level scheduler that consists of three decision makers [13]. The first scheduling level is responsible for the actual placement of the VMs in a given data center. The second level decides in which data center of a given CSP a VM is to be placed. The third level – which is not yet applied – decides on a cloud federation level to which federated partner a given VM is to be migrated.

Simulation Software In order to examine grid and cloud phenomena, researchers rely on simulations because real world testbeds are complex to configure and very expensive. Since grid computing came in the research focus, a lot of simulation frameworks have been implemented (e.g. [5, 6, 12, 16]). Many of the original grid simulators are now reused and extended for cloud computing, but there are also completely new simulators for special purposes.

A frequently used, event-driven simulation framework is *CloudSim* [6]. It simulates configurable components of a distributed cloud like data centers, physical and virtual machines and workloads. As CloudSim is versatile and extendible, it was often used and extended in cloud research (e.g. [5]); Devi et al. give a comprehensive overview [8]. CloudSim is also the basis of FederatedCloudSim which supports the simulation of cloud federations. The idea of FederatedCloudSim was developed in the research project *EASI-CLOUDS* [21]. Its concept was described in [13], a first description of the implemented framework was given in [14]. Figure 1 shows a cloud scenario with components that can be simulated using FederatedCloudSim.

In the next section we will describe further extensions to the FederatedCloudSim framework which include the support of SLA management, VM migration between data centers, processing of real-world workload traces and event logging.

2. FederatedCloudSim Extensions

Based on the work presented in [14], FederatedCloudSim was extended with respect to SLAs and the simulation of real-world workloads. Such workload traces are provided by a Dutch cloud service provider named Bitbrains [19]. Previously to our work, they have only been statistically examined [18]. In comparison to grid workload traces which mostly include scientific applications, these trace logs contain the resource usage of business-critical VMs like database, web and financial application servers. In order to process the traces, a new parser and utilization model were implemented in FederatedCloudSim.

The new version also supports migrations of virtual machines between data centers. The FederatedDatacenterBroker (short: DCBroker) can trigger migrations based on its strategy. Migrations can be considered necessary if SLOs are violated or close to being violated. Such events are detected by the SLA manager of the CSP which checks all SLOs for every active VM at regular intervals. The SLO thresholds and the length of the intervals can be configured in the XML configuration file. The SLA manager reports every (imminent) SLA breach to the VM scheduler of the respective data center.

The SLAs implemented are guaranteed resource quotas and availability or uptime. As example resources we consider CPU and RAM in this paper and allow CSPs to overprovision these resources. For simplicity, virtual machines can be created using predefined *SLA categories*, for example *bronze*, *silver* and *gold*. Each category then defines the amount of requested resources (here, CPU and RAM), the agreed availability, the limit for the number of migrations and the cost for VM execution. The gold category, for instance, could require that the VM is never or rarely migrated and that it is available 99.9% of the time.

If a problem cannot be solved at the data center level, the local VM scheduler can decide to transfer the VM to a different data center of the same CSP. For this, it contacts the FederatedDatacenterBroker. If there is another data center offering enough resources for the respective VM, then the VM is transferred to the new data center. Otherwise the VM is paused until one of the data centers is ready to schedule the VM again on local resources. Downtimes like this and other SLA violations are measured and taken into account in the billing process as possible fines have to be paid to the customer. VM migrations also impact the availability because it is necessary to copy at least the RAM contents. If a VM is migrated to a different data center, the copy operation usually also includes virtual disks so that the

downtime is increased. The exact downtime lengths can be configured.

Finally, logging capabilities of the framework were expanded to allow a deeper analysis of the simulation results.

3. SLAs and Strategies

Service Level Agreements In the experiments we consider the following SLAs:

- Uptime and migrations. We assume that the uptime of an application running in a VM is reduced by migrations of this VM. In our simple model migrations within a data center reduce the uptime by five minutes, migrations in between data centers by ten minutes. Uptime can also be affected by crashing machines, but we do not simulate those yet.
- RAM and CPU. RAM and CPU are often exclusively assigned to VMs, but since many VMs only use a fraction of the RAM / CPU assigned, these resources can also be overprovisioned by the CSP. If this is done, the scheduler has to make sure that the system always provides enough resources to the VMs. RAM and CPU are only examples; similar SLAs include storage throughput or network bandwidth.

Scheduling Strategies The current implementation of the framework includes several scheduling strategies. The strategies that we use in our experiments are combinations of strategies for selecting the VM or VMs to migrate and the hosts where they are to be placed. We have implemented some of the strategies suggested in [3]. For the selection of the VMs we use:

- Minimization of Migrations (M): Like the other strategies, this one is triggered when at least one of the SLA thresholds is exceeded. It migrates the least number of VMs necessary to fall below the threshold(s) and tries to be as close to the threshold(s) as possible.
- Maximization of Migrations (X): This strategy also migrates VMs, until the value in question is below the threshold. Yet, it chooses the smallest VMs with respect to utilization and thus maximizes the number of VMs migrated. The idea is that these VMs can be more easily placed within in the same data center.
- Highest Potential Growth (H): This strategy chooses the VMs that have the lowest utilization relative to the allocated resources. The idea is that these VMs have the highest potential growth and can therefore cause new threshold violations.

The strategies for the selection of the hosts are:

- Best Fit (B): This strategy finds the host that has the lowest resource consumption and can provide the VM with the requested resources while staying below the RAM and CPU threshold.

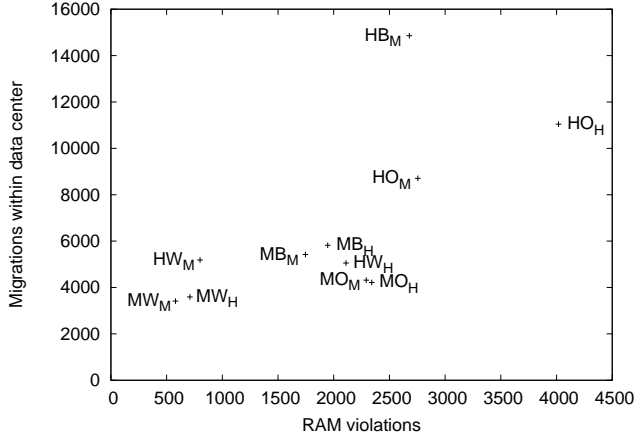


Figure 2. Trace 1: Migrations within data center and RAM violations caused by strategy combinations; index M (H) indicates that the DCBroker uses Minimization of Migrations (Highest Potential Growth); outlier HB_H (3843, 25966)

- Worst Fit (W): This strategy finds the host that can provide the VM with the requested resources while being the closest to the RAM and CPU threshold.
- Least Overprovisioning (O): This strategy makes its decisions based on the requested resources of the VMs. The *overprovisioning factor* is defined as the sum of the requested resources divided by the total amount of resources. Then this strategy chooses the host with the smallest overprovisioning factor for RAM and CPU resources.

Additionally, there are strategies for the DCBroker which decides whether to migrate VMs between the data centers. Here we use the two strategies from [3] again:

- Minimization of Migrations (M-DC): If there is no space left in the data center, VMs are migrated to other data centers, and again this strategy chooses as few VMs as possible to reach values below the threshold(s).
- Highest Potential Growth (H-DC): see above.

The SLA manager also considers load thresholds for every host. If a given resource (RAM or CPU) is used $\geq 80\%$ then this host is flagged as potentially in danger. The SLA manager waits for one more simulation ticks (5 minutes each) and checks whether the load is still $\geq 80\%$. If so, the local scheduler is called to reschedule the VMs of this host. If the load drops below this threshold, no action is performed. If the load of a given resource suddenly rises above 90%, then an immediate reschedule is triggered.

Simulated Systems We use the Bitbrains traces [19] mentioned in Section 2 for our simulations. The original data center configuration of the hosts that the workload was recorded on is undisclosed; so we used a configuration that was created in a way that we have enough resources to provide the VMs with RAM and CPU capacity in average load

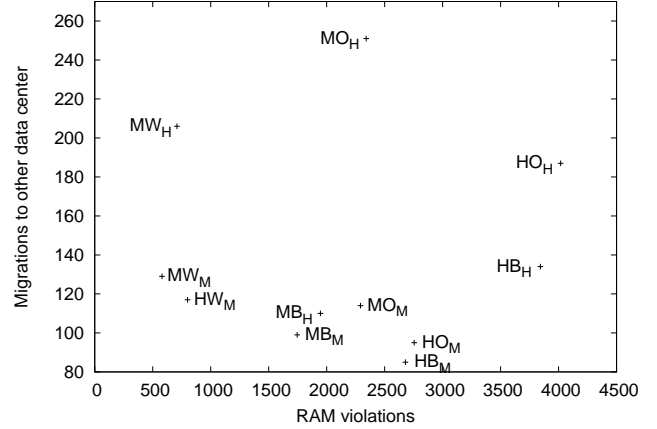


Figure 3. Trace 1: Migrations to other data centers and RAM violations caused by strategy combinations; outlier HW_H (2110, 326) not displayed

situations. As the resource consumption varies throughout the trace, we see a lot of migrations in the data center.

The data centers for the traces are configured as follows:

Trace 1: The data center has 33 hosts each equipped with 36 cores @ 2,9 GHz and 32 GB RAM. Hence, in total there are 1188 cores and 1056 GB RAM. 1250 VMs are run on these nodes for one month.

Trace 2: The data center has 10 hosts each equipped with 30 cores @ 2926 MHz and 40960 MB RAM. In total they sum up to: 300 cores and 400 GB RAM. 500 VMs are run for three months.

In each simulation we use a second data center that has a lot of spare resources so that all VMs that are transferred to this data center can be executed. Initially there are no VMs in the second data center. Our analysis focuses on in the first data center and regards the second one only as a dump for VMs migrated out of the first one.

4. Evaluation

For the evaluation of the (combinations of) strategies we have run simulations with both traces and analyzed the logs regarding broken SLAs (availability, RAM, CPU) and the number of migrations within and out of the data center. Some of the results are plotted in the Figures 2, 3, 4 and 5. Figure 2 and 4 rank the strategies regarding RAM violations and migrations within the data center for trace 1 and 2, respectively. Figure 3 and 5 show the same, but now for migrations out of the data center.

The first noticeable thing is that there is actually a strategy in Figure 2, namely the combination of *Minimization of Migration* and *Worst Fit* (MW), that is optimal with respect to both, migrations and RAM violations. This appears odd because one would expect that either RAM violations or migrations are high because migrations are used to prevent RAM violations. However, this can be explained by another metric, namely the number of migrations to other data cen-

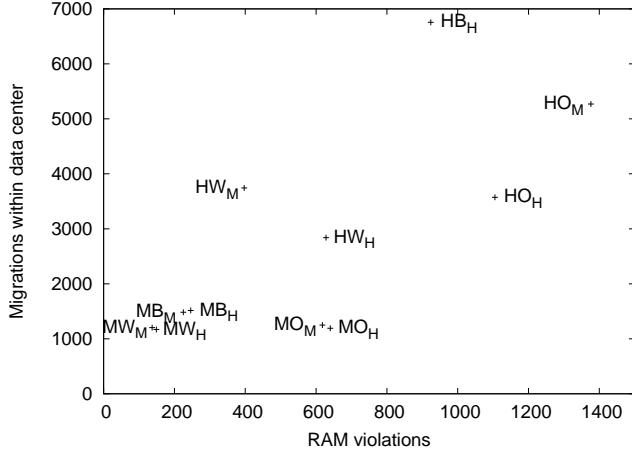


Figure 4. Trace 2: Migrations within data center and RAM violations caused by strategy combinations; outlier HB_M (658, 10994) not displayed

ters, which is relatively high for MW and all other strategies in the lower left corner of Figure 2. By migrating VMs to another data center the overall resource consumption in that data center decreases so that the remaining VMs can effectively share more local resources which leads to fewer migrations and resource violations. The same can be observed in Figure 4 and 5.

Comparing the strategies with each other, it sticks out that there is clear hierarchy of the DCBroker strategies for the two traces. Minimization of Migrations is almost always better than Highest Potential Growth, and – not depicted in the plots – Maximization of Migrations is embarrassingly worse than the others. It is not surprising that the number of migrations is higher for the latter because they tend to pick VMs that are small in size and therefore have to migrate more of them. The size of the virtual machines is also the (likely) explanation for the number of RAM violations because, while the number of VMs migrated to the other data center is higher, the total size of the VMs is usually lower. The latter two strategies have therefore less space to arrange the remaining VMs in the local data center. In this respect the plots are deceptive because the “bad” strategies actually utilize more resources than Minimization of Migrations.

As one can see in the figures, the traces differ significantly in the number of migrations and SLA violations, but this is mainly caused by the sizes of the systems. Other than that, there are many similarities in the plots with respect to the quality of the strategies. Generally, Worst Fit tends to break less SLAs than Best Fit. The reason is that Worst Fit leaves the VMs more legroom so that they are less likely to violate SLAs. This waste of space has no negative consequences because if VMs do not longer fit, they are simply migrated to another one. So, we can observe the disadvantage of Worst Fit in Figure 3 and 5 which show that it usually results in a higher number of migrations out of the data center. The strat-

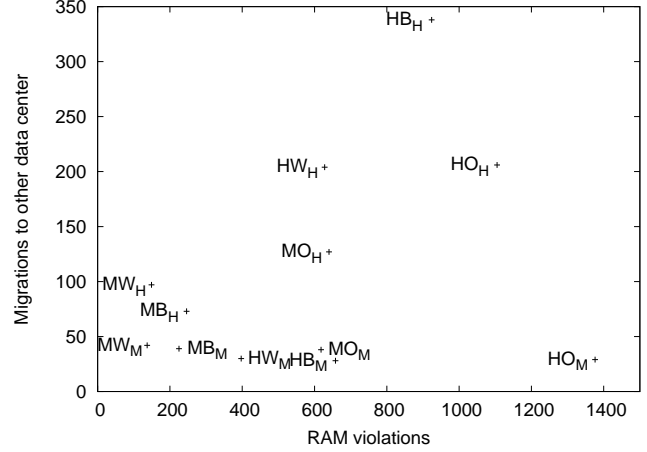


Figure 5. Trace 2: Migrations to other data centers and RAM violations caused by strategy combinations

egy Least Overprovisioning, which focuses on the resources requested rather than the resources currently used, performs badly for the two traces. It is designed on the assumption that requested resources are also eventually used, but in real clouds this might not be the case due to predetermined VM size categories or simply inaccurate estimations.

The same argument can be applied for VM selection where Highest Potential Growth is usually (much) worse than Minimization of Migrations. Here again, the potential growth rarely happens, and it is better to use a strategy which considers the current state of the VM.

For lack of space we cannot display all results. We picked migrations and RAM violations instead of availability and CPU violations, but the outcomes are very similar. The number of migrations and availability are closely connected where the former is a little bit more precise than the latter. Migrations because of CPU violations are included in the plots; we could have shown them instead of RAM violations or the sum of the two and got essentially the same plots.

5. Conclusion

In this paper we have presented different strategies for SLA-based scheduling in cloud data centers. The strategies were simulated using the FederatedCloudSim framework which has been extended to allow for SLA management and monitoring, simulation of real-world cloud traces and resource overprovisioning. We compared the strategies with regard to the number of migrations and the number of CPU and RAM violations. In our simulations, the combination of *Minimization of Migrations* for VM selection and *Worst Fit* for host selection shows the best results regarding SLA breaches. In some respect, this is specific to our scenario in which we allow migrations to another data center so that strategies might actually benefit from wasting space. Yet, the analysis also shows that strategies which concentrate on the current state and resource usage generally perform better than

strategies that prepare for the potential usage. We conclude that there is a lot of low fluctuation of the resource usage in real-world data centers, but VMs are nevertheless often overdimensioned so that they never use all the requested resources. For this reason, overprovisioning is feasible if, at the same time, the actual usage is monitored and if in case of imminent SLA violations VMs are rescheduled sooner rather than later.

In the future, FederatedCloudSim will be extended by adding more scheduling strategies for migrating VMs within and in between data centers. Our main goal, however, is to add another level of scheduling / brokering and to analyze strategies in cloud federations. In the simulation, federated cloud service providers will be able to conclude SLAs with each other and offer services to their partners. For the analysis, we will also add a fine-grained and configurable cost model so that one can investigate the impact of strategies on the CSP's revenue.

References

- [1] A.-F. Antonescu, P. Robinson, and T. Braun. Dynamic sla management with forecasting using multi-objective optimization. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 457–463. IEEE, 2013.
- [2] A. Beloglazov and R. Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13):1397–1420, 2012. ISSN 1532-0634. . URL <http://dx.doi.org/10.1002/cpe.1867>.
- [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. . URL <http://dx.doi.org/10.1016/j.future.2011.04.017>.
- [4] N. Bobroff, A. Kochut, and K. Beaty. Dynamic placement of virtual machines for managing sla violations. In *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*, pages 119–128. IEEE, 2007.
- [5] R. Buyya, R. Ranjan, and R. Calheiros. Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. *Algorithms and architectures for parallel processing*, pages 13–31, 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] M. Comuzzi, C. Kotsokalis, C. Rathfelder, W. Theilmann, U. Winkler, and G. Zacco. A framework for multi-level sla management. In *Service-Oriented Computing. IC-SOC/ServiceWave 2009 Workshops*, pages 187–196. Springer, 2010.
- [8] R. K. Devi and S. Sujana. A survey on application of cloudsims toolkit in cloud computing. *International Journal of Innovative Research in Science, Engineering and Technology*, 3, 2014.
- [9] D. Durkee. Why cloud computing will never be free. *Commun. ACM*, 53(5):62–69, May 2010. ISSN 0001-0782. . URL <http://doi.acm.org/10.1145/1735223.1735242>.
- [10] V. C. Emeakaroha, M. A. Netto, R. N. Calheiros, I. Brandic, R. Buyya, and C. A. De Rose. Towards autonomic detection of sla violations in cloud infrastructures. *Future Generation Computer Systems*, 28(7):1017–1029, 2012.
- [11] F. Jrad, J. Tao, and A. Streit. Sla based service brokering in intercloud environments, 2012.
- [12] D. Kliazovich, P. Bouvry, and S. U. Khan. Greencloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, 62(3):1263–1283, 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. Federated-cloudsim: a sla-aware federated cloud simulation framework. In *Proceedings of the 2nd International Workshop on Cross-Cloud Systems, CCB@Middleware 2014, Bordeaux, France, December 8, 2014*, pages 3:1–3:5, 2014.
- [15] K. Lu, R. Yahyapour, P. Wieder, C. Kotsokalis, E. Yaqub, and A. I. Jehangiri. Qos-aware vm placement in multi-domain service level agreements scenarios. In *Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*, pages 661–668. IEEE, 2013.
- [16] A. Núñez, J. L. Vázquez-Poletti, A. C. Caminero, G. G. Castañé, J. Carretero, and I. M. Llorente. iCanCloud: A Flexible and Scalable Cloud Infrastructure Simulator. *Journal of Grid Computing*, 10(1):185–209, 2012.
- [17] R. Sakellariou and V. Yarmolenko. Job scheduling on the grid: Towards sla-based scheduling. In *High Performance Computing Workshop*, pages 207–222, 2006.
- [18] S. Shen, V. van Beek, and A. Iosup. Statistical characterization of business-critical workloads hosted in cloud datacenters. In *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*, pages 465–474, May 2015. .
- [19] The Grid Workloads Archive. The grid workloads archive, <http://gwa.ewi.tudelft.nl/>, Jan. 2016. URL <http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains>.
- [20] H. N. Van, F. D. Tran, and J.-M. Menaud. Sla-aware virtual resource management for cloud infrastructures. In *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on*, volume 1, pages 357–362. IEEE, 2009.
- [21] 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.