SLA amongst Users and Providers in Multi-Cloud Environment Through Negotiation Model

https://doi.org/10.3991/ijes.v9i2.22151

Merita Kasa Halili (✉)
South East European University, Tetovo, North Macedonia
mk2635@seeu.edu.mk

Betim Cico
Metropolitan Tirana University, Tirana, Albania

Abstract—Cloud Computing conducts application, infrastructure services or platform to a very large number of users with more choices and continuous changing requirements. Cloud providers are occupied in organizing data warehouses to arrange the continuous growth in cloud user's acceptance. Features of cloud computing services have afforded an important tendency of companies choosing these services. In this case, many cloud users, who intend a certain service, and many cloud providers, who provision those services, create a competitive market. Cloud computing is an architecture that provides various computing resources as service over the internet. The provision of such computing resources over the internet must be scalable and should be provisioned rapidly on-demand. This service provisioning in the cloud computing system is based on the Service Level Agreements (SLAs). The SLA represents service contracts signed between the service consumer and cloud service provider. In cases of violation of the SLA, penalties are associated. SLA also works as an assurance for the service consumer/client for ensuring efficient utilization of available resources to minimize the cost of resource provisioning. To reduce/avoid SLA violations in the cloud computing system, we have proposed an SLA reduction framework in which we have considered three steps: (a) Scheduling algorithm to efficiently allocate cloudlets to the virtual machines based on the processing time of Host. (b) MinVm Scheduling algorithm: Allocates cloudlets to virtual machines based on cloudlet allocation counts to each VM (Virtual Machine). (c) Credit-Based VM Migration algorithm uses the credit of the VMs to take VM migration. To analyze the performance of the proposed framework and to compare results with existing SLA aware scheduling algorithm cloudsim, simulation tool was used. At the end we have shown the results with the help of the graphs. Our work presents an approach, based on analysis and comparison of minimum utilization policy and novel SLA policy. It indicates the SLA variations and number of Virtual Machine migration. Moreover, we conduct experiments for comparing the effectiveness of the proposed utility-based solution. Through the tailored schemes and composition of our algorithm, these results achieve to depict very less SLA violation.

Keywords—SLA; SLA violation; Cloud Computing; Cloudsimulator

1 Introduction

In Cloud Computing, Service Level Agreement (SLA) represents a contract that points out synthesis of QoS (Quality of Services) in level and type in connection of cloud provider and cloud user. Recently, cloud computing is getting a very important software delivery paradigm, mostly for matters of increased scalability. The scalability advantages are proficient by the ability of autonomously and elastically scaling up or down so that cloud users' preferences (SLAs) can be satisfied [1]. There are a lot of reasons in using Cloud computing resources, mostly, because major tech companies use the opportunity to afford on-demand absolute space of resources for an absolute time of execution, while holding all the investment dangers on their own, and as a consequence charge you in pay-per-use mode. Amongst others, the cloud computing services are as well proposed as Infrastructure as a Service (IaaS) where the cloud provider delivers a virtualized computing infrastructure, as a service. In this kind of service most of the administration and monitoring responsibilities are the provider's concern [2]. The aim of an SLA is to crossover the gap between cloud service provider and cloud users or customers. However, there are a lot of issues and unsolved problems regarding the specification and the quantification of SLAs.

Problem statement: Nevertheless, yet there is a gap in formalizing Service Level Agreements (SLAs) between the cloud provider and user. In addition, the industry and academia communities have not placed a boundary through a formal model or formula which advices a certain business whether to opt for a cloud IaaS facility or not. Another subject that has to deal with SLA is the SLA violation. During the analysis and research regarding this problem we identify some mechanisms and techniques proposed by various researchers that help in the early management and detection of the SLA violation.

2 Literature Review

Recently, various approaches have been proposed for solving the SLA violation and failure problems that occur. Ivan Breskovic [3], introduces a method for autonomic ally deriving public SLA templates based on user requirements. The SLA mapping approach uses public SLA templates to show group of products traded on the market, and private SLA templates to depict user requirements. After a new public SLA template is obtained, users' SLA mappings represented to the first public SLA template are autonomous adjusted in order to be feasible to the new public SLA template. In addition, the first public SLA templates can be erased and substituted by the new templates, abstaining continuously growing number of public templates on the market, and establishing the market more efficiently. However, the authors do not specify the main parameters of SLA configuration in multi-cloud environment. In our framework we have indicated the energy consumption and Virtual machine migration roles in the negotiation model. Atul Gohad [4] presents an approach towards reaching self-adaptive cloud cooperation. Their model opts the best possible cloud cooperation between multiple potential dependent cloud provider collaboration links. The for-

mation of cloud provider cooperation is widely based on cloud provider possibilities, tenancy requirements, cost modeling of every cloud provider and its functional capability at the Software as a Service (SaaS) layer. They present an algorithm to form dynamic cloud cooperation and thus define the most appropriate connections within the providers. The system architecture consists of these components: Cloud Provider Model, Power Consumption Tracker, provider cost analyzes and the Provider Cost Graph. However, this research lack on Virtual Machine Scheduling optimization that we have enhanced in our paper. A. Kertesz [5] propose a novel holistic architecture called SLA-based Service Virtualization (SSV) made on agreement negotiation, brokering and service deployment combined with business-oriented operation. They also use the Autonomic Computing technology to inspect their introduced architecture and study how the principles of Autonomic Computing emerge in the basic components of the architecture in order to face with changing user requirements and on-demand failure handling. Autonomic systems need high-level guidance from humans and decide which steps should be done to maintain a stable system. This kind of systems constantly adapts themselves to changing environmental conditions. Additionally, to this negotiation model we have provided alternatives of combining VMs (Virtual Machines) to deal with certain solution in Cloudlet, Data Center and VM specifications.

Shyam S. Wagle [6] propose the SLA insured brokering framework which matches the requirements of the clients with SLA provided by Cloud Service Providers (CSPs) using similarity matching algorithm and readiness to pay capacity for the services. It also calculates the services certified by CSPs for certifying and ranking the CSPs. Initially, they establish the cloud brokering architecture which matches the clients' requirement. Clients' requirements are matched using SLA templates provided by CSPs and provides the service package based on their requirement and readiness to pay capacity of them for the services. Second, instead of proposing SLA based services, their framework also measures the current services offered by CSPs and compares with SLA proposed by corresponding CSPs for CSP certification and ranking the CSPs. In comparison to the Shyam research, our algorithm. The proposed algorithm achieves better results (VM migration: 188 & host shutdown: 85) in Cloudsim. Abdel-Rahman Al-Ghuwairi [7] proposed a flexible model to bring up-to-date cloud computing SLA time to time, in order to obtain a dynamic SLA complying with conditions and eliminate costly violations. The Architecture of their example is Service Oriented Architecture (SOA). Their model contains the services listed below: Monitor, analyzer and terms generator. The monitor constantly controls the QoS parameters like accessibility and the data that will be transmitted to the analyzer to determine if there has been a violation as opposed to the values approved on the first SLA. In case of a violation in any of the QoS [8] parameter occurs, deliverance to customer and providers will be stored in the database with its parameters. In case an update is required as approved on the first SLA or the contract has expired and needs to be renewed, the new terms generator offers other parameters for each QoS in accordance to the SLA-violation and SLA-penalty implementation which stored in SLA database. The above evaluated papers lack of expressing the heterogeneity of services and communication between geo-distant clouds by avoiding SLA-violation. However, this

research lacks the analysis of main QoS parameters like: Energy consumption in cloud servers and Virtual machine migration.

The research goal: The aim of an SLA is to crossover the gap between cloud service provider and cloud users or customers. However, there are a lot of issues and unsolved problems regarding the specification and the quantification of SLAs. To reduce/avoid SLA violations in the cloud computing system, in our work we propose an SLA reduction framework in which we have considered three steps: Migration control of VMs, Energy efficiency and VmScheduling [9]. To analyze the performance of the proposed framework and to compare results with existing SLA aware scheduling algorithm CloudSim simulation tool was used. At the end we have shown the results with the help of the graphs. Our work presents an approach, based on analysis and comparison of minimum utilization policy and novel SLA policy. It indicates the SLA variations and number of Virtual Machine migration. Moreover, we conduct experiments for comparing the effectiveness of the proposed utility-based solution. Through the tailored schemes and composition of our algorithm, these results achieve to depict very less SLA violation.

3 Cloudsim Toolkit

Cloud computing frequently qualifies issues of distinct load, energy performance, designing and scheduling other applications and services for the cloud environment that can be resolved by utilizing CloudSim. CloudSim can be explained as an independent platform that ensures an extendable simulation toolkit that affords modeling and simulation of cloud computing systems and application providing environments. CloudSim consists of these fundamental assets which makes it suitable for simulating cloud:

1. Ensures controlled and a repeatable environment to establish and simulate cloud entities
2. Can be expanded to comprise user defined policies for cloud
3. Ensures variable infrastructure modeling, variable network architecture and federated cloud support
4. Ensures VM provisioning, Host provisioning, network provisioning and application provisioning [10].

3.1 Cloudsim features

i. Datacenter

The datacenter is the most important element of any cloud computing system. Datacenter is comprised of hosts and hosts arrange virtual machines that are responsible for lower level or current processing of cloudlets (tasks) in the system. Overall, a data center is a physical environment solution considered for housing computer systems and related components. The required physical environment facility encompasses power supplies with the opportunity to provide backup power [11]. CloudSim Data

center class represents a cloud resource, whose host list are virtualized. It deals with processing of VM queries (handling of VMs) instead of processing cloudlet related queries [10].

ii. Datacenter Broker

This class models a broker, which is engaged for mediating between cloud users and cloud service providers depending on users' QoS requirements. The broker disseminates service tasks through clouds [13]. Datacenter Broker works as a mediate between user and cloud service provider, whose main role is to ensure SLA and QoS requirements specified by the user. Datacenter Broker contains the submit Cloudlets method that schedules cloudlets to VMs [14].

iii. Host

The host manages a list of VMs. The host assigns processing capabilities to VMs like speed, MIPS, memory, storage.

iv. Virtual Machine

Virtual machines are portable, inter-operable, and logically independent. Each VM established is assigned to a Host where all its computational requirements are met. A virtual machine is a multiuser distributed resource that provides heterogeneous service to all computer or network resources [15].

v. Cloudlet

Cloudlet in CloudSim represents the workload, which should be executed along the simulation run of the CloudSim simulation engine. Cloudlet in CloudSim is a model class that is within the package. In CloudSim, cloudlets are computational requirements which VMs should fulfill. Overall cloudlets are tasks in CloudSim [16].

4 Experimental Setup

This section contains an analysis and justification of scientific results. The division into units is recommended.

In this section we have provided the execution parts of the algorithm. We have described the content of the executed files, their responsibility, and the alternatives of combining VMs to deal with certain solution.

The tables shown below indicate the characteristics of the various components used in the simulation.

Table 1. VM Specification

Parameter	Description
Size	7000 MB
Ram	512 MB
MIPS	600+ rn.nextInt(300)
Bandwidth	1000 bps
Number of Pes	1
Vmm	Xen

Table 2. Cloudlet Specification

Parameter	Description
Length	700+ rn.nextInt(300)
File size	300 MB
Output size	300 MB
Number of Pes	1

Table 3. DC (Data Center) Specifications

Parameter	Description
Architecture	X86
Os	Linux
Vmm	Xen
Time zone	10.0$ //time zone this resource located
Cost	3.0 $ // the cost of using processing this resource
CostPerMem	0.05$ //the cost of using memory in this resource
costPerStorage	0.1$ //the cost of using storage in this resource
costPerBw	0.1$ //the cost of using Bw in this resource

Table 4. DC (Data Center) Specifications

Parameter	Description
Ram	2048 MB
Storage	1000000
Bandwidth	10000 bps

4.1 Cloudsim simulation setup

Datacenterbroker.java: Package: org.cloudbus.cloudsim in the source folder.

This file is responsible for scheduling cloudlets to VMs so that user SLA and QoS are satisfied.

DatacenterBroker.java file contains the submitCloudlets method which consists of the scheduling algorithm.

The scheduling algorithm used here is minVmSelection algorithm. This algorithm selects the minimum allocated VM for allocation based on the allocation history of that VM.

a) CloudSimExample6.java
b) Package: org.cloudbus.cloudsim.examples in the example folder.

Using this file, we can create a scalable simulation in CloudSim. This file contains methods responsible for

- Creating VMs
- Creating Cloudlets
- Creating Datacenter
- Creating DatacenterBroker
- And the method for printing the results on the console.

c) PowerVmSelectionPolicy

Enterprises are moving to PowerVM virtualization to depict multiple workloads onto fewer systems, by improving server utilization, and by that reducing cost. This technology provides a secure and scalable virtualization environment for heterogeneous applications (AIX, IBM, Linux) built in accordance to the advanced reliability, availability, and serviceability methods and the advanced characteristics of the Power systems platform. As a summary, we can define the PowerVM as the industrial strength virtualization solution System servers and blades. Through this technology clients control costs and improve overall performance, flexibility, availability and energy efficiency.

Package: org.cloudbus.cloudsim.power

This method is responsible for identifying VMs to be migrated.

d) PowerVmAllocationPolicy

Package: org.cloudbus.cloudsim.power in the source folder.

This file contains a method for identifying over-utilized hosts. For running the simulation, we need to combine VmAllocationPolicy and VmSelectionPolicy.

VmAllocationPolicy determines overloaded hosts and VmSelectionPolicy migrates Vms from overloaded hosts.

e) LrMu.java

Package: org.cloudbus.cloudsim.examples.power.random in example folder.

LrMu file does the work of combining VmAllocationPolicy and VmSelectionPolicy

5 Changing Parameters

As Cloud computing has become the new paradigm for task execution, Virtual Machine configuration indicates a critical role in the performance of this paradigm. Furthermore, it depicts the process of selecting the equivalent configuration for task execution by conveying some static and dynamic factors of tasks. In our case study, the parameters of scheduling can be set using the following files:

Case 1: Changing number of VMs, the configuration of VMs, Hosts

SLA Constants: CloudSim is one of the major breakthroughs, that have been widely employed by the research community to publish their optimized results related to their proposed algorithms and parameters regarding the VM migration. In our case, constant values are conveyed in constants.java file under org.cloudbus.cloudsim.examples.power package.

The number of VMs and Number of Hosts: Host executes actions related to management of VM in regard to creation and destruction. It has a defined policy or rules for provisioning memory and allocation policy. A host is associated to a data-center which can host virtual machines.

The RandomConstants java file under org.cloudbus.cloudsim.examples.power.randompackage can determine the number of VMs and Number of Hosts.

6 Simulation and Result Analysis

Simulation Parameters:

Table 5. VM Specifications

Parameter	Description
VM Type	Small Instance: 1 EC2 Compute Unit, 1.7 GB
VM MIPS	{2500, 2000, 1000, 500}
VM PES	{1, 1, 1, 1}
VM Ram	{1, 1, 1, 1}
VM Bandwidth	100 Mbit/s
VM Size	2.5 GB

Table 6. Host Specifications

Parameter	Description
Host Type	HP ProLiant ML110 G5 (1 x [Xeon 3075 2660 MHz, 2 cores], 4GB)
Host MIPS	{1860, 2660}
Host PES	{1, 1, 1, 1}
Host Ram	{4096, 4096}
Host Bandwidth	1 Gbit/s
Hos Storage	1 GB

Constants:
Number of Hosts: 800
Number of VMs: 1052

6.1 Result comparison of VM migration and host shutdown

The following result depicts the number of VM migration and host shutdown. The proposed algorithm achieves better results (VM migration: 188 & host shutdown: 85) compared to the existing algorithm (VM migration: 209 & host shutdown: 87).

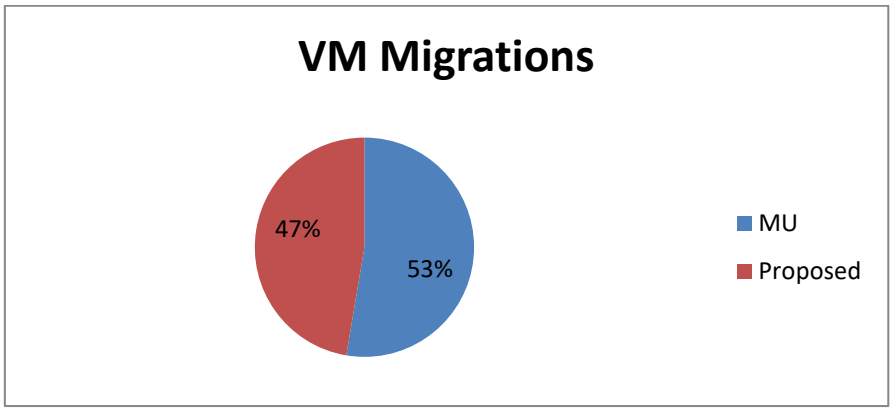

Figure1. Surface of errors of Sugeno type fuzzy systems

Fig. 1. The number of VM (Virtual Machine) migration and host shutdown

Package: org.cloudbus.cloudsim.examples.power.random

SIMULATION_LIMIT = 60 * 60

With existing VM selection policy (PowerVmSelectionPolicyMinimumUtilization)

```
Experiment name: random_lr_mu_1.2
Nr. of hosts: 50
Nr. of VMs: 50
Total simulation time: 3600.00 sec
Energy consumption: 1.73 kWh
Nr of VM migrations: 209
SLA: 0.02918%
SLA perf degradation due to migration: 0.27%
SLA time per active host: 10.75%
Overall SLA violation: 1.87%
Average SLA violation: 11.03%
Nr. of host shutdowns: 87
Mean time before a host shutdown: 676.25 sec
StDev time before a host shutdown: 668.17 sec
Mean time before a VM migration: 19.38 sec
StDev time before a VM migration: 8.16 sec
Exec. time - VM selection mean: 0.00009 sec
Exec. time - VM selection stDev: 0.00030 sec
Exec. time - host selection mean: 0.00600 sec
Exec. time - host selection stDev: 0.01957 sec
Exec. time - VM reallocation mean: 0.00145 sec
Exec. time - VM reallocation stDev: 0.00104 sec
Exec. time - total mean: 0.01309 sec
```

```
Exec. time - total stDev: 0.02141 sec
```

With Proposed VmSelection Policy:

```
Experiment name: random_lr_mmt_1.2
Nr. of hosts: 50
Nr. of VMs: 50
Total simulation time: 3600.00 sec
Energy consumption: 1.67 kWh
Nr. of VM migrations: 188
SLA: 0.01836%
SLA perf degradation due to migration: 0.23%
SLA time per active host: 7.89%
Overall SLA violation: 0.76%
Average SLA violation: 9.49%
Number of host shutdowns: 85
Mean time before a host shutdown: 598.29 sec
StDev time before a host shutdown: 545.31 sec
Mean time before a VM migration: 17.67 sec
StDev time before a VM migration: 8.00 sec
Exec. time - VM selection mean: 0.00009 sec
Exec. time - VM selection stDev: 0.00030 sec
Exec. time - host selection mean: 0.00436 sec
Exec. time - host selection stDev: 0.01414 sec
Exec. time - VM reallocation mean: 0.00173 sec
Exec. time - VM reallocation stDev: 0.00135 sec
Exec. time - total mean: 0.01155 sec
Exec. time - total stDev: 0.01618 sec
```

6.2 Result comparison of energy consumption efficiency

Energy consumption, or we can also refer to energy efficiency, indicates the complete infrastructure to reduce functional costs while maintaining vital QoS. Whereas energy optimization can be achieved by combining resources as per the ongoing utilization, efficient and effective virtual network roadmap and thermal situation of computing hardware and related nodes. Shown in the graph in Figure 7, the proposed algorithm consumes less energy (1.67 kWh) compared to the existing (MU) algorithm (1.73), meaning that in long distance usage it leads to energy optimization and by that be energy consumption efficient.

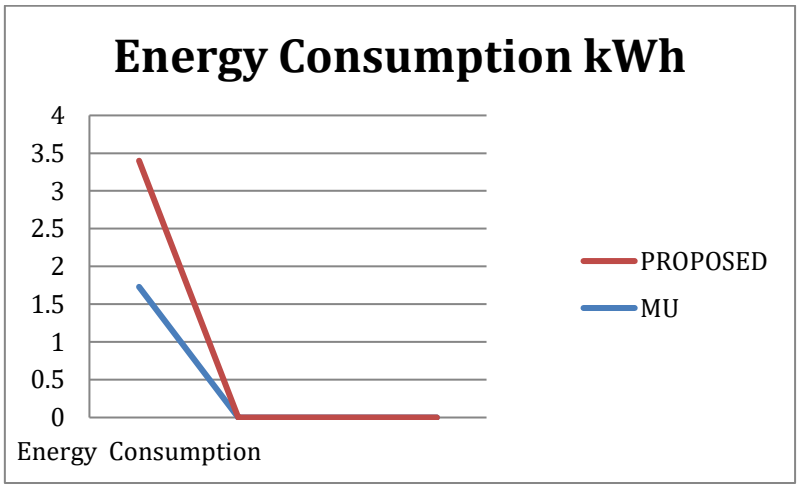

Fig. 2. Energy consumption for MU and proposed algorithm

7 Conclusion

In this paper, we presented an SLA violation reduction negotiation model in which we have considered three steps: Migration control of VMs, Energy efficiency and Virtual Machine Scheduling. We introduced our experimental setup implemented in CloudSim where we have provided the execution parts of the algorithm with the associated parameters and results. We have described the content of the executed files, their responsibility, and the alternatives of combining VMs to deal with certain solution. Furthermore, we have proposed a more comprehensive VM selection policy to deal with energy consumption, VM migration and host shutdowns.

8 References

[1] Rola Motawie, Mahmoud M. El-Khouly1, M. Samir Abou El-Seoud. "Security Problems in Cloud Computing". iJES – Volume 4, Issue 4, 2016. https://doi.org/10.3991/ijes.v4i4.6538

[2] M.K. Halili, B. Cico. "Towards Custom Tailored SLA in IaaS environment through negotiation model". 7th Mediterranean Conference on Embedded Computing MECO'2018, Budva, Montenegro. https://doi.org/10.1109/meco.2018.8406005

[3] Iva Breskovic et, al. "Cost-Efficient Utilization of Public SLA Templates in Autonomic Cloud Markets". 2011 Fourth IEEE International Conference on Utility and Cloud Computing 978-0-7695-4592-9/11 $26.00 © 2011 IEEE DOI 10.1109/UCC.2011.38 229. https://doi.org/10.1109/ucc.2011.38

[4] Atul Gohad et, al. "Towards Self-adaptive Cloud Collaborations". 2013 IEEE International Conference on Cloud Engineering. https://doi.org/10.1109/ic2e.2013.11

[5] A. Kertesz a, G. Kecskemeti a, I. Brandic b. "An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous Cloud environments". Future Generation Computer Systems 32 (2014) 54–68. https://doi.org/10.1016/j.future.2012.05.016

[6] Shyam S. Wagle. “SLA Assured Brokering (SAB) and CSP Certification in Cloud Computing”. 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing. https://doi.org/10.1109/ucc.2014.167
[7] Abdel-Rahman Al-Ghuwairi et, al. “A Dynamic Model for automatic Updating cloud computing SLA (DSLA)”. ICC ’16, March 22-23, 2016, Cambridge, United Kingdom.
[8] N. Lakki, J. Oubaha et, al. “The Effect of the Multi-Objective Dynamic Metric on QoS and the Energy in Network Manets”, International Journal of Recent Contributions from Engineering, Science & IT, Vol 7, No 2, 2019, eISSN: 2197-8581. https://doi.org/10.3991/ijes.v7i2.10439
[9] M. Kasa Halili, B.Cico, “Sla Management For Comprehensive Virtual Machine Migration Considering Scheduling And Load Balancing Algorithm In Cloud Data Centers”, International Journal on Information Technologies & Security, № x, 201x, Varna, Bulgaria.
[10] D. Bulaja, K. Bozic et, al.” Introduction to Cloudsim”. International Scientific Conference on Information Technology and Data Related Research. Sinteza 2019. https://doi.org/10.15308/sinteza-2019-189-194
[11] Rashid, MD. Abdur. “Data Center Owerview”. National Academy for planning and Development. Volume 28. 2019. ISSN: 1607- 8373.
[12] H. Amipara. “A Survey on CloudSim Toolkit for Implementing Cloud Infrastructure”. IJSTE - International Journal of Science Technology & Engineering | Volume 1 | Issue 12 | June 2015.ISSN (online): 2349-784X.
[13] T. Chaterjee. V.K. Ojha, et, al. “Design and Implementation of an Improved Datacenter-Broker Policy to Improve the QoS of a Cloud. P. Krömer et al. (eds.), Proceedings of the Fifth Intern. Conf. on Innov. In Bio-Inspired Comput. and Appl. IBICA 2014, Advances in Intelligent Systems and Computing 303,281, © Springer International Publishing Switzerland 2014. https://doi.org/10.1007/978-3-319-08156-4_28
[14] Driss Riane & Ahmed Ettalbi “Efficient and Optimal Service Component Distribution across Multiple Clouds”. iJES – Vol. 6, No. 3, 2018. https://doi.org/10.3991/ijes.v6i3.9567
[15] Chatterjee, Tamojit & Ojha, Varun & Adhikari, Mainak & Banerjee, Sourav & Biswas, Utpal & Snasel, Vaclav. (2014). Design and Implementation of an Improved Datacenter Broker Policy to Improve the QoS of a Cloud. https://doi.org/10.1007/978-3-319-08156-4_28
[16] Mishra, Suchintan & Sahoo, Manmath. (2017). On using CloudSim as a Cloud Simulator: The Manual. 10.13140/RG.2.2.30215.91041.

9 Authors

Merita Kasa Halili – PhD candidate at South East European University, Faculty of Contemporary Sciences and Technology and Teaching Assistant at State University of Tetova at the department of informatics. Merita is awarded a Google scholarship in 2012 and also the President’s award “golden engineering ring”.

Betim Cico - full professor in Metropolitan Tirana University, Tirana, Albania; research interest in fields of Cloud computing, Digital System Design, Digital Image Processing, Advance Computer Architecture, Internet of Things, FPGA. He is author of many research papers, projects and supervisor of several PhD students. bcico@umt.edu.al

Article submitted 2021-02-18. Resubmitted 2021-03-26. Final acceptance 2021-03-26. Final version published as submitted by the authors.