Evaluation of traffic-aware VM placement policies in distributed Cloud using CloudSim

Raja Benali[‡], Hana Teyeb[*§], Ali Balma[†], Samir Tata[*¶] and Nejib Ben Hadj-Alouane[‡]
[¶]Almaden Research Center, IBM Research, San Jose, CA, USA
[*]Institut TELECOM, TELECOM SudParis, CNRS UMR Samovar, Evry, France
Email: hana.teyeb@telecom-sudparis.eu, samir.tata@it-sudparis.eu
[§] University of Tunis El Manar, Faculty of Sciences of Tunis
[†] University of Tunis, National Higher Engineering School of Tunis, Tunisia
Email: ali.balma@gmail.com
[‡]University of Tunis El Manar, National Engineering School of Tunis-OASIS, Tunisia
Email: raja.benali1991@gmail.com, nejib_bha@yahoo.com

***Abstract*—Due to the complexity of managing virtual machines arrival and departure in a geographic distributed Cloud, we tackle the dynamic placement and migrations of VMs with simulation techniques. In this work, we present the study of VMs placement policies in geographically distributed data centers (DCs) using the cloud simulator CloudSim. We have extended the simulator to take into consideration the inter-DCs communication and migration. Through simulations, we show an example of a VMs placement and migration approach and discuss the obtained results.**

I. Introduction

Cloud Computing has emerged as new paradigm where resources are delivered on a pay-as-you-go basis. Tenants may benefit from many cloud services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [1]. In recent years, the number of companies moving into the cloud has also increased considerably [2]. In order to serve world-wide end-users, modern cloud data centers (DCs) are located in different geographic locations and generally connected by an IP over wavelength-division-multiplexing (WDM) network [3], [4].

With the rise of the popularity of Cloud Computing services, the number of applications having high demand on networking resources has increased significantly. In the IaaS, which is the focus of this paper, tenants can benefit from on-demand provisioning of computing, storage and networking resources [1]. Some of the emerging cloud-based applications include web hosting, video streaming and real time applications. Each type of application has different composition, configuration and deployment requirements.

In this context, many management issues still deserve additional investigations.Virtual machine (VM) placement is one of the other real problems that many Cloud owners face. Evaluating the performance and the efficiency of placement policies in real cloud environment under different conditions and for different applications and service models is challenging. In fact, the use of real cloud environment has several limitations such as the system size and configuration which make the reproduction of some results a very difficult task. Moreover, with a real cloud system, the evaluation of some critical scenarios and failures is not supported. In addition, the access to real cloud environment requires payments.

To cope with these limitations, there are some simulation tools that offer the possibility of evaluating different placement, allocation and scheduling policies under different conditions. These tools, offer a controllable environment free of cost that mimic the behavior of a real cloud environment. In such an environment, users may test their models within a critical situation in order to study the overheads of their models and to cope with performance degradation before deploying in real world [5].

The purpose of this paper is to evaluate some placement policies that concern the assignment of networking VMs to the different DCs. To achieve this objective, we have to simulate geographically distributed DCs connected with a network topology. In this work, we extend the cloud simulator CloudSim [6] to take into consideration the communication and the migration of VMs between the different DCs. We study an example of VM placement and migration policies based on optimization models that have been presented in previous works [7], [8]. We present and discuss different simulations results conducted on the different optimization models in order to evaluate the effectiveness and the limitations of the proposed placement policies.

The remainder of this paper is organized as follows: In Section II we present a survey of literature and we describe our contribution. As for the Section III, it presents a background on the CloudSim tool. In Section IV we present the simulated cloud environment and the different extensions and modifications that we have conducted on the CloudSim tool. Section V discusses the simulation results. Finally, we conclude in Section VI.

II. Related work

In this paper we aim to simulate geographically distributed DCs connected with a network topology. Our objective is to evaluate some placement policies that concern the assignment of networking VMs to the different DCs. To achieve this goal,

978-1-5090-1663-1/16 $31.00 © 2016 IEEE
DOI 10.1109/WETICE.2016.29

we need to execute our experiments for a period of time that can be quite long. Therefore, we need to pay Cloud provider such for resources we use during development and evaluation processes. Due to these reasons, simulation has been chosen as the right environment in which we can experiment without high cost. It gives better results with a low cost of learning. There are some works that have proposed simulation environments in order to evaluate the VMs placement and resource allocation policies [9], [10]. Several Grid simulators, presented below, have been proposed for modeling Grid based environments. Among these simulators, we mention:

- SimGrid [11]: is a scientific instrument to study the behavior of large-scale distributed systems such as Grids, Clouds, HPC or P2P systems. One of the main issues of GridSim is the scalability problem that impedes the simulation of large-scale systems.
- GreenCloud [12]: is an extension to the network simulator NS2 [9] developed for the study of cloud computing environments. The drawback of the GreenCloud Simulator is that it confines its scalability to only small DCs.
- iCanCloud [13]: is a software simulation framework for large storage networks. The main disadvantage of iCanCloud is that it models and simulates only EC2 (Elastic Compute Cloud 2) environments.

In addition, those toolkits present some limitations such as the incapability to isolate the multi-layer service abstractions (IaaS, PaaS, and SaaS) required by the Cloud computing environments. None of the current distributed system simulators offer the environment that can be directly used for modeling cloud environments. Nevertheless, CloudSim, which is generalized, and extensible simulation framework, allows seamless modeling, simulation, and experimentation of emerging cloud computing infrastructures and application services. By using CloudSim, we can test the performance of a newly developed application service in a controlled and easy to set-up environment. The vast features of CloudSim would speed up the development of new application provisioning algorithms for Cloud Computing. However, some aspects of distributed environment are not implemented. We cite for example the inter-DCs VM migration and communication.

In the next section, we present some background knowledge about the CloudSim tool.

III. Background

In this section we will present the CloudSim toolkit and some other extensions based on this simulator. Then, we will present relevant existent works that have used CloudSim for simulations.

A. CloudSim

CloudSim is an open source simulator which enables seamless modeling, simulation, and experimentation of cloud computing and application services. It was developed in the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the Computer Science and Software Engineering Department of the University of Melbourne. The CloudSim toolkit supports system and behavior modeling of cloud system components such as DCs, VMs and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited efforts. CloudSim helps the researchers and developers to focus on specific system design issues without getting concerned about the low level details related to cloud-based infrastructures and services.

B. CloudSim Extensions

In this section we present some existent CloudSim extensions.

CloudAnalyst presented in [14] is a CloudSim-based Visual Modeler developed to simulate large-scale Cloud applications with the purpose of studying the behavior of such applications under various deployment configurations. In [15], the authors have extended the cloud simulation platform CloudSim by adding the file striping and data replica management function. In [16], the authors have addressed some limitations of CloudSim such as the lack of links between DCs and the no possibility to create a VM in more DCs. They have proposed a ring topology to allow the sharing of information and services between different DCs. They have improved the method of creating VMs to allow the creation of VMs in several DC, which improves fault tolerance in this type of environment. In [17], five task scheduling algorithms are proposed and implemented based on CloudSim platform.

C. Simulation using CloudSim

In this section, we present some existent works that have used the CloudSim toolkit, in order to evaluate some placement, allocation or scheduling policies in cloud environment.

In [18], the authors have conducted competitive analysis on the single VM migration and dynamic VM consolidation problems in DCs. However, the authors have not taken into account the possibility of VMs migration between DCs. They have evaluated the proposed algorithms by extensive simulations using the CloudSim toolkit. In [19], the authors have proposed a VM placement and migration approach to minimize the data transfer time consumption. As for [20], it presents an algorithm that improves communication performance by reducing the overall traffic cost of VMs. In [21], the authors have proposed a distributed dynamic VM consolidation approach that improves the energy efficiency in cloud environments. They have validated their approach using the CloudSim toolkit and a real world PlanetLab workload. In [22], the authors have presented an algorithm that minimizes the VM migration time as well as the number of migrations in order to avoid performance degradation encountered during the migration process. In [23], the authors have demonstrated that the migration control in energy aware VM consolidation not only saves energy but also shows an improvement in VM migration number which reduces the network traffic. Moreover, the simulations were carried out using CloudSim toolkit for one day workload trace of Planetlab. In [24] the authors have proposed an algorithm that aims to reduce the power consumption and minimize the SLA violations.

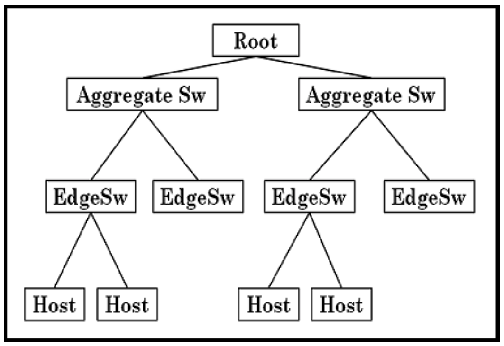

Fig. 1. Data center network topology "Tree".

The aim of this paper is to simulate, evaluate and validate VMs placement and migration policies within a geographically distributed cloud infrastructure.

In the next section, we will present an introduction to the CloudSim architecture and the extensions that we have implemented.

IV. Simulated Cloud Configuration

In this section we will present first, the cloud architecture that we have used to evaluate the different VM placement and migration policies. Then, we will present the implementation of the inter-DCs migration and communication extension.

A. Problem Description

We aim to model a geographically distributed DCs connected by a network. The different placement policies that we are evaluating concern the assignment of VMs to the different DCs while considering the following constraints:

- *Capacity constraint*: the set of VMs assigned to a specific DC must not exceed the capacity of the DC in terms of physical resources (RAM, CPU, Storage),
- *Location constraint*: it restricts the placement of VMs in a particular number of DCs,
- *Allocation constraint*: it ensures that each VM is running on only one DC.

The objective of these placement policies is to minimize the overall backbone network traffic generated by the communication of the different VMs.

1) Cloud architecture: CloudSim supports the simulation of inter-DCs network topology using a BRITE file [25] for modeling link bandwidth and associated latency. It will help us in simulating the communication over the backbone network. It is also useful for the network traffic monitoring under different scenarios. In order to simulate and evaluate the amount of traffic circulating in each DC, we have implemented Tree [26] network topology which is widely used because of its simplicity in terms of reducing costs. We have implemented this topology using CloudSim classes *RootSwitch*, *AggregateSwitch* and *EdgeSwitch* as shown in figure 1.

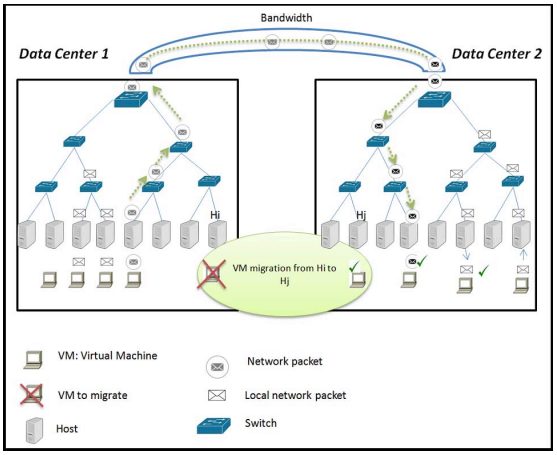

Fig. 2. Communication and migration between DCs.

B. VM placement policies

The placement policies that we are evaluating concern only the placement of VMs in DCs. However, each VM needs to be assigned to a specific host as well. In this paper, we evaluate the performance of two specific VMs placement policies: static and dynamic approaches.

The different placement polices used in the simulations are presented as follows:

- *VM/DC placement policy*: it is based on Integer Linear Programming (ILP) optimization models. It was first presented in [8]. The aim of this model is to minimize the overall traffic between DCs. However, the proposed formulation does not take into consideration the network topology of DCs and the experiments presented are not suitable for a realistic cloud environment.
- *VM/Host placement policy*: to assign a VM to a specific host, a VM Monitor uses provisioning policies for allocating resources to VMs. The main functionality of those policies is to select available host in a DC, which meets the memory, storage, and availability requirement for a VM deployment [27]. CloudSim implements two types of VM allocation policies: i) a time-shared policy which assigns cores to VMs on demand, and ii) space-shared policy that dynamically distributes the capacity of a cores among VMs. In this paper we have used these two allocation policies to study their impact combined with the proposed optimization models.

C. VM communication model

The CloudSim toolkit supports the communication between VMs within the same DC. We have extended the existent classes to take into consideration the communication of VMs among distributed DCs. We have used the network topology described above to perform the data transfer. By extending the CloudSim class "*AppCloudlet*", we have implemented a new generic type of application that allows for a certain number of VMs to communicate between each other. The VMs may

TABLE I
NUMBER OF HOSTS PER DC

Data center	DC1	DC2	DC3	DC4	DC5	DC6
# Hosts	280	240	400	280	240	360

TABLE II
HOST CHARACTERISTICS

Characteristics	SA	OS	Mips	RAM	Storage	pes
Value	x86	Linux	2660	4096 MB	1 GB	2

be placed in the same DC or in a distant one. The amount of traffic exchanged between VMs varies from 10 to 100 MBps.

D. VM migration policy

In this paper, we are also evaluating a dynamic VM placement policy that uses VM migration technique in order to reduce the overall backbone network traffic that includes the communication and the migration traffic. This optimization model is an extension of the static model presented in [8]. However, the migration decisions include the VM/DC mapping only.

CloudSim implements several VM allocation policies that dynamically optimizes the VM allocation using VM migration. Some policies aim to minimize the number of SLA violations due to the overload of hosts.

However, these policies are applicable for the intra-DC VM migration i.e. the migration of a VM from a host to another within the same DC. Hence, we have integrated a VM migration method that aims to perform the migration of a VM from a DC source to another DC destination.

Indeed, VM migration consists of transferring a VM from a source to a destination. Basically there are two types of VM migration: hot (or live) and cold migration [28]. In "live" migration the VM is still running during the migration process and the user does not perceive any changes. As for the "cold" migration, the VM loses its status and the user notices a service interruption.

Unfortunately, the last version of CloudSim does not support the VM live migration. Hence, we have extended the existent "cold" migration method to perform the inter-DC VM migration as follows.

1) Shutting down the VM to be migrated and all its running services,
2) Deallocate the VM from the DC source,
3) Create the VM in the DC destination using CloudSim allocation policies,
4) Resume the VM and its services.

In the next section, we will present and discuss the different simulation results conducted on the optimization placement and migration policies.

V. SIMULATION

In this section, we study the performance of the two implemented placement algorithms: static and dynamic.

TABLE III
HARDWARE METRICS FOR INSTANCE TYPES

	CPU(core)	Memory(GB)	Storage(GB)
Small	1	1.7	1*160
Medium	1	3.75	1*410
Large	2	7.5	2*420

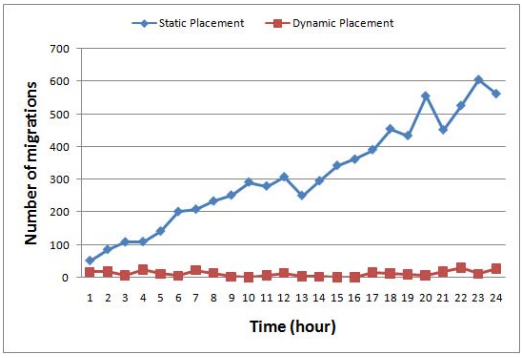

Fig. 3. Variation of the number of migrations.

A. Modeling the CloudSim extension

Figure 2 describes the extensions achieved in CloudSim simulator to perform distant communication. The network topology between DCs allows the possibility of sending packets from a DC source to a distant one within the available bandwidth, as well as the possibility of migrating VMs between DCs.

To perform communication between VMs within a geographically distributed DCs, we have implemented a personalized class "*NetworkApp*", that extends "*AppCloudLet*" class. Given a traffic matrix and a number of VMs, it will build an application composed of networking VMs. We have also extended the "*RootSwitch*" class to support sending and receiving packets between distant DCs.

The VM migration between DCs was implemented by adding new DC tag, "*VM_NETWORK_MIGRATION*" that is used by the DC source to inform the DC destination about the migration request. It will also send the description of the migrated VM.

B. Use Case

In this section, we present a simulation use case that uses the implemented extension. Then, we evaluate static and dynamic VM placement policies.

The different simulations were carried out on a machine that has an Intel Xeon 3, 3Ghz CPU and 12 GB of RAM. We have used the commercial solver CPLEX 12.5 [29] to solve the different ILP formulations.

Without loss of generality and for the sake of presentation, we have fixed the number of DCs to six. The capacity of a DC is directly related to the number of hosts. The table I presents the number of hosts per DC. The host characteristics are presented in the table II. We assume that VMs have an

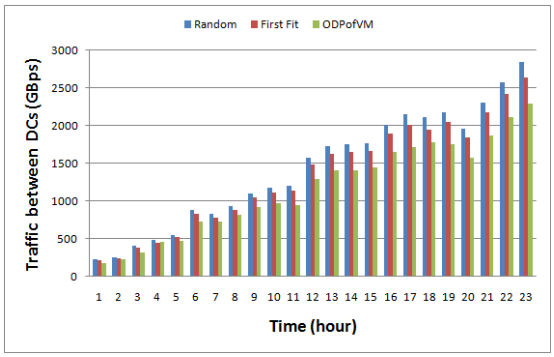

Fig. 4. Placement algorithms comparison: 1VM per 60s.

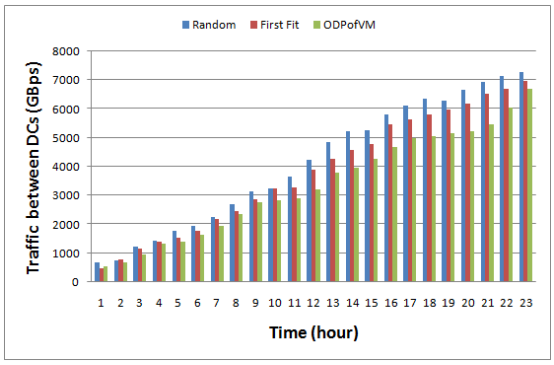

Fig. 5. Placement algorithms comparison: 1VM per 20s.

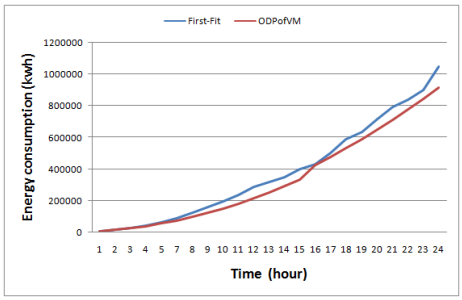

Fig. 6. Energy comparison graph.

instance type (Small, Medium and large) that are presented in table III. The different values of hardware metrics are provided by Amazon Elastic Computing Cloud (EC2) [30]. We assume that each VM can be placed in two possible DCs, known a priory, but eventually, each VM is assigned to only one DC.

In all simulations, we have considered that the VMs arrival follows the Poisson distribution [31]. The simulation duration was fixed to one day i.e. 24 hours and the results were plotted every hour.

First, we discuss the effectiveness of the two algorithms in term of the number of migrations. The graph presented in figure 3, shows the variation of the number of migrations for the static and the dynamic placement policies. In contrast to the static placement algorithm, the dynamic algorithm

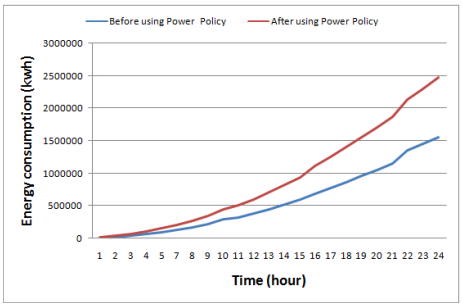

Fig. 7. Power policies impact in energy consumption.

(ODPofVM) considers both communication and migration traffic while making placement and migration decisions. In fact, we have re-executed the static placement algorithm at the beginning of each iteration and we have compared the results with the dynamic version. We may conclude that the dynamic policy prevents from performing excessive migrations. Hence, it may be more suitable for a realistic cloud environment as it maintains the system stability.

In order to validate the effectiveness of the dynamic placement algorithm, we have conducted a comparison with two baseline algorithms: First-fit and Random. The graph presented in figures 4 and 5 demonstrates that the Optimal Dynamic Placement of VM (ODPofVM) algorithm gives the best placement plan in terms of minimizing the traffic between DCs. As mentioned previously, the VMs arrival follows the Poisson distribution. We have varied the mean number of VMs arrival per second from 1 VMs per 60 second to 1 VM per 20 second. The results are depicted in figures 4 and 5.

The DC energy consumption has an important impact in cloud environment. Thus, we have compared the energy consumption between the First-fit and the ODPofVM placement algorithms. As shown in figure 6, we note that the ODPofVM algorithm decreases the DCs energy consumption compared to the First-fit algorithm.

CloudSim implements network-aware and energy-aware placement policies. As we aim to reduce the network traffic, we have combined those two placement policies to simulate and evaluate its impact on the energy consumption. Figure 7 presents the energy consumption before and after including power-aware policies.

We have also studied the impact of the two types of VM allocation policies offered by CloudSim: time-shared policy and space-shared policy. Figure 8 and 9 present the energy consumption and the total data transfer delay combined with the optimized placement policies.

VI. Conclusion and Future Work

In this paper, we have presented an evaluation study of some VM placement and migration policies. To simulate a distributed cloud environment, we have used CloudSim toolkit. We have implemented some extensions in order to take into consideration several aspects such as the VMs communication between different DCs and the inter-DCs migration. We

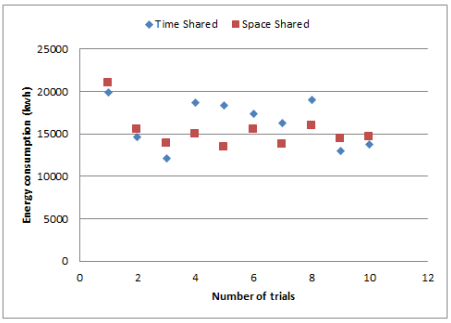

Fig. 8. Energy comparison between Time-Shared and Space-Shared policies.

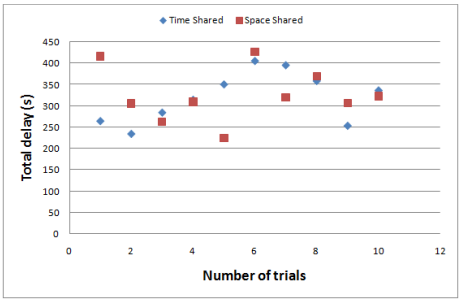

Fig. 9. Delay comparison between Time-Shared and Space-Shared policies.

have combined some existent CloudSim allocation policies with ILP-based optimization policies. We have also compared the placement policies with random and first-fit algorithms. Through extensive simulations, we have shown the effectiveness of the optimization policies in terms of minimizing the backbone traffic. As a future work, we will perform a wide set of experiments showing the impact of different types of applications and its impact in the placement decisions.

REFERENCES

[1] P. M. Mell and T. Grance, "Sp 800-145. the nist definition of cloud computing," Tech. Rep., 2011.

[2] C. I. Forum, "UK cloud adoption and trends for 2013," Tech. Rep., 2012. [Online]. Available: http://www.cloudindustryforum.org/white-papers/uk-cloud-adoption-and-trends-for-2013

[3] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers." ACM, 2009.

[4] K. Church, A. Greenberg, and J. Hamilton, "On delivering embarrassingly distributed cloud services," *Hotnets VII*, 2008.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities." IEEE, 2009.

[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*, 2011.

[7] H. Teyeb, A. Balma, N. B. Hadj-Alouane, and S. Tata, "Optimal virtual machine placement in large-scale cloud systems," in *IEEE 7th International Conference on Cloud Computing, Anchorage, AK, USA, June 27 - July 2, 2014*. IEEE, 2014.

[8] H. Teyeb, A. Balma, N. Ben Hadj-Alouane, S. Tata, and A. B. Hadj-Alouane, "Traffic-aware virtual machine placement in geographically distributed clouds," in *Control, Decision and Information Technologies (CoDIT), 2014 International Conference on*. IEEE, 2014, pp. 024–029.

[9] D. Kliazovich, P. Bouvry, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," *The Journal of Supercomputing*, 2012.

[10] F. Messina, G. Pappalardo, D. Rosaci, and G. M. Sarne, "An agent based architecture for vm software tracking in cloud federations," in *Complex, Intelligent and Software Intensive Systems (CISIS), 2014 Eighth International Conference on*. IEEE, 2014, pp. 463–468.

[11] H. Casanova, "Simgrid: A toolkit for the simulation of application scheduling," in *Cluster computing and the grid, 2001. proceedings. first ieee/acm international symposium on*. IEEE, 2001.

[12] P.-N. Clauss, M. Stillwell, and Genaud, "Single node on-line simulation of mpi applications with smpi." IEEE, 2011.

[13] T. Issariyakul and E. Hossain, *Introduction to network simulator NS2*. Springer Science & Business Media, 2011.

[14] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, 2010.

[15] S. Long and Y. Zhao, "A toolkit for modeling and simulating cloud data storage: An extension to cloudsim." IEEE, 2012.

[16] G. Belalem and S. Limam, "An extension and improvement of the cloudsim simulator," *International Journal of Digital Information and Wireless Communications (IJDIWC)*, 2011.

[17] Z. T. He, X. Q. Zhang, H. X. Zhang, and Z. W. Xu, "Study on new task scheduling strategy in cloud computing environment based on the simulator cloudsim," in *Advanced Materials Research*. Trans Tech Publ, 2013.

[18] 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*, 2012.

[19] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*. IEEE, 2010.

[20] H. T. Vu and S. Hwang, "A traffic and power-aware algorithm for virtual machine placement in cloud data center," *International Journal of Grid & Distributed Computing*, 2014.

[21] S. S. Masoumzadeh and H. Hlavacs, "Integrating vm selection criteria in distributed dynamic vm consolidation using fuzzy q-learning."

[22] R. S. TsepoMofolo, "Heuristic based resource allocation using virtual machine migration: a cloud computing perspective," *International Refereed Journal of Engineering and Science (IRJES)*, 2013.

[23] M. A. H. Monil, R. Qasim, and R. M. Rahman, "Incorporating migration control in vm selection strategies to enhance performance." *IJWA*, 2014.

[24] R. Sinha, N. Purohit, and H. Diwanji, "Power aware live migration for data centers in cloud using dynamic threshold," 2011.

[25] A. Medina, A. Lakhina, I. Matta, and J. Byers, "Brite: An approach to universal topology generation." IEEE, 2001.

[26] A. Headquarters, "Cisco data center infrastructure 2.5 design guide," 2007.

[27] D. N. A. K. Ms Jayshri Damodar Pagare. (2015) Design and simulate cloud computing environment using cloudsim. ijcta.

[28] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "Improving virtual machine migration in federated cloud environments," in *Evolving Internet (INTERNET), 2010 Second International Conference on*. IEEE, 2010.

[29] I. ILOG, "Inc. cplex 12.5 user manual," 2012.

[30] E. Amazon, "Amazon elastic compute cloud," *Retrieved March*, 2010.

[31] F. A. Haight, "Handbook of the poisson distribution," Tech. Rep., 1967.