Acta Scientific COMPUTER SCIENCES

Volume 3 Issue 5 May 2021

Research Article

Characterization and Simulation of Static Cloud Behaviors Using CloudSim

ChukwuNonso H Nwokoye[1]*, Chioma Okeke[1], Obiajulu Ositanwosu[1] and Peace O Ezzeh[2]

[1]*Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria*

[2]*Computer Education Department, Federal College of Education (Technical), Asaba*

***Corresponding Author:** Chukwu Nonso H Nwokoye, Department of Computer Science, Nnamdi Azikiwe University, Awka, Nigeria.

Received: April 13, 2021

Published: April 26, 2021

© All rights are reserved by **Chukwu Nonso H Nwokoye., *et al*.**

Abstract

In this paper, we sought to perform simulation for cloud services using CloudSim. Recently, CloudSim has been revered for its suitability for cloud simulation purposes and availability i.e. easy download and usage by research students for academic projects. Specifically, two specific cloud scenarios were designed and implemented using the simulator. Several services such as data center, data center broker and virtual machines' allocation policies i.e. time-shared and space-shared were modeled and used for simulation studies. Actually, sensitivity analyses involved tweaking length of instructions and input file size. Several results were generated and attendant conclusions were presented.

Keywords: CloudSim; Cloud Services; Datacenter; Length of Instructions

Introduction

Recently, cloud computing has generated a plethora of immense benefits. Most of all is the conception of making computing a utility and allowing the commercialization of software, platform and infrastructure. The implication is that it can be not only bought (by users), but can also be sold (by cloud providers). This idea has eliminated the burden on companies/developers, i.e. the anxiety and fretting concerning the absence of infrastructures or frameworks required for the operational successes of projects. As Armbrust., *et al*. [1] puts it, "The infrastructure can be leased in a cloud provider, which handle different demands of workloads and prevent resources from remaining idle or overloaded". In the words of "Providers of cloud computing have a robust infrastructure that consists of servers, switches, routers, air conditioning devices, structured cabling, power supplies, energy, and other. Cloud computing systems involve different types of devices and are large-scale systems, which makes the prototyping of this magnitude quite difficult". Universally, in cloud researches, simulation is become an accepted method of assessing system behaviors and for the validation of schemes such as scheduling, costs and efficiency. Researchers worldwide have used simulation to create different ways of evaluating the behavior of the systems and validating strategies for task scheduling, cost models, scaling with energy efficiency, etc. Herein, We perform simulation for cloud services using the CloudSim toolkit. This is to highlight the impact of length of instructions, input file size and two allocation policies. Note that CloudSim [3] is a tool for modeling and simulation of cloud computing environments, with focus on strategies for CPU scheduling, allocation of virtual machines, optimization of energy consumption, among other.

The paper is organized as follows: Section II is the review of pertinent literature, Section III is where we performed the modeling and simulation of cloud scenarios/behaviours, Section IV contains the results of the study and Section V contains the conclusion and proposal for future studies.

Related works

Several authors have reviewed the strengths of CloudSim. Buyya., *et al*. [4] confirmed the advantages of CloudSim and they include; "support for modeling and simulation of large scale cloud

computing infrastructure, including data centers on a single physical computing node; and a self-contained platform for modeling data centers, service brokers, scheduling, and allocations policies". The benefits of CloudSim was also brought to the fore by a survey done by Fakhfakh., *et al*. [5] and a comparative study performed by Tian., *et al* [6]. Some studies that employed CloudSim for investigations are as follows. Beloglazov., *et al*. [7] propose an architecture that provides efficient energy consumption in cloud environments; it uses policies and scheduling algorithms to reduce energy consumption while maintaining the service level agreements contracted. In Guérout., *et al*. [8], CloudSim is customized to provide data on energy efficiency, such as consumption in KWh, percentage of broken contracts, percentage of CPU usage, among others. Goutam., *et al*. [9] present techniques for dynamic load balancing for cloud computing environments. Using CloudSim, the proposed implementation called RAM Broker decides, in a proactive manner, in which cloud resource the workload should be allocated. Fittkau., *et al*. [10] report on an experience to provide a perspective of the client in the simulations, introducing enhancements in CloudSim. It is possible to compare different cloud computing providers and select the best features for the client requirements. Souza., *et al*. [2] proposed an approach where the client entity is based on several characteristics that affect the performance of a cloud computing system, including different modes of submission and their behaviour when the waiting time between requests (think time) is considered.

Methodology

Modelling and simulation

The simulation toolkit used is CloudSim version 3.0.3 developed by developed in the CLOUDS Laboratory at the University of Melborne, Australia. To be able to use this package, we also downloaded the Eclipse integrated development environment (IDE) for Java Developers, the Apache Common Math version 3.6.1 and the Java software. These software were all installed on a 64 bit Windows operating system. Figure 1 shows the Eclipse IDE where the modeling and simulation is done.

The algorithm (pseudocode) for the simulation is as follows:

- Import utilities: ArrayList, Calendar, LinkedList, List, Random
- Import org.cloudbus.cloudsim: Cloudlet, CloudletScheduler, CloudletSchedulerSpaceShared, CloudletSchedulerTimeShared, Datacenter, DatacenterBroker, DatacenterCharacteristics, Host, Log, Pe, Storage, UtilizationModelFull, Vm, VmAllocationPolicySimple, CloudSim, PeList, BwProvisionerSimple, PeProvisionerSimple, RamProvisionerSimple
- Initialize the CloudSim package before creating any entities.
- Create Datacenter(s) (Datacentercharacteristics, HostList, Processing element List)
- Create Broker
- Create Cloudlets (Defines the workload i.e. cloudletLength, pesNumber, cloudletFileSize, cloudletOutputSiz)
- Create VMs (Define the procedure for Task scheduling algorithm) and specify these characteristics diskSize, ram, mips, bandwidth, vCPU and VMM)
- Starts the simulation (automated process, handled through discrete event simulation engine)
- Print results when simulation is over (Outputs).

Figure 1: Eclipse IDE workspace.

Using the installed software, the impact of the length of instructions using the two types of allocation policy i.e. time-shared and space-shared was evaluated. The choice of length of instructions is due to the fact that this parameter concerns the users and their activities. The following cloud scenarios were used to x-ray the execution time of cloudlets.

Scenario 1

We coded a datacenter, a datacenter broker, 40 cloudlets (tasks/workloads) and 10 virtual machines. Specifically, in this study a datacenter has 4 hosts with the following processing elements: 4

cores of 1000 mips each, 8GB RAM, 100GB storage (100000 MB) and 1 mbps (8000 kbits/s) network bandwidth. The 40 cloudlets has the following attributes: 4000 length of instructions, 300 kb input file size, 400 kb file size, and 1 core CPU and utilization model to full. The study also included 10 virtual machines with the following characteristics: 20GB storage disk, 2GB RAM, 1 vCPU (1000 mips CPU speed), 1000 kbit/s bandwidth and the timeshared scheduler for cloudlets execution. Figure 2 and 3 are depicts the first and second 20 cloudlets for scenario 1.

Figure 2: First 20 cloudlets for scenario 1.

Figure 3: Second 20 cloudlets for scenario 1.

Scenario 2

The parameters of scenario 1 was maintained except for length of instructions and input file size, which were increased to 100,000 and 3000, respectively. Figure 4 and 5 are depicts the first and second 20 cloudlets for scenario 2.

Scenario 3

The parameters of scenario 1 were kept constant but the allocation policy was changed from time-shared to space-shared. Figure 6 and 7 are depicts the first and second 20 cloudlets for scenario 3.

Figure 4: First 20 cloudlets for scenario 2.

Figure 5: Second 20 cloudlets for scenario 2.

Scenario 4

The parameters of scenario 1 were kept constant but length of instruction length of instructions and input file size, which were increased to 100,000 and 3000, respectively. Additionally, the allocation policy was changed from time-shared to space-shared. Figure 8 and 9 are depicts the first and second 20 cloudlets for scenario 3.

Results and Discussions

This section contains the results of the simulation done for the aforementioned scenarios. Discussions of what the results imply are placed after the results below.

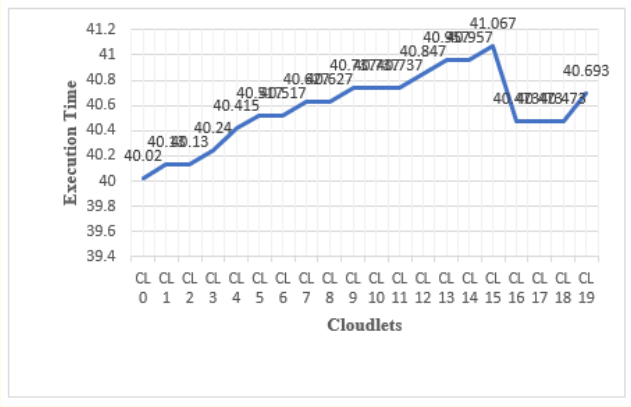

Figure 6: First 20 cloudlets for scenario 3.

Figure 7: First 20 cloudlets for scenario 3.

Figure 8: First 20 cloudlets for scenario 4.

Figure 9: First 20 cloudlets for scenario 4.

Note that the simulation was done using the parameters of scenario 1 described above. Other scenarios (2, 3 and 4) are slight adjustments made to scenario 1 in order to highlight the impact of length of instructions, input file size and two allocation policies.

For the first and second 20 cloudlets depicted as Figure 2, one could see that the execution time (i.e. the actual CPU time) revolved around 80 until it got to cloudlet 15, where it increased to 81.74 from 80.524. Subsequently, it increased to 120.103 at cloudlet 16. Then it increased to 121.023 at cloudlet 26 and evolved around 121 till the last cloudlet 39.

With the increase of the length of instructions as well as the input file size (Figure 3 and Figure 4), execution time started at 200.248 for cloudlet 0 until cloudlet 11 where the execution time increased to 201.036 at cloudlet 11. Subsequently, at cloudlet 16 it increased to 300.205 and revolved around 300 till cloudlet 26. At cloudlet 27, the execution time moved to 301.084 and stayed a 301 till the end (at cloudlet 39). It is noteworthy to say that figures 2 to 5 showed highlighted the effect of the time-shared allocation policy.

On the impact of space-shared allocation policy depicted as figures 6 to 9, one could notice a dramatic reduction in the execution time, with length of instruction and input file size placed at 40000 and 300 respectively. It started at 40.02 for cloudlet 0, increased to 41.067 for cloudlet 15 and went back to 40.473 for cloudlet 16. It revolved around 40 for the rest of the simulation. Comparing figure 2 and figure 3 with figure 6 and figure 7, one can see the trend is not exactly the same, more so, with drop in execution time.

For Figure 8 (first 20 cloudlets) and Figure 9 (second 20 cloudlets), which are results of an increased length of instructions (100,000) and input file size (3000), execution time started at 100.054 for cloudlet 0 and increased to 101.041 at cloudlet 14/15. The execution time revolved around 100 till cloudlet 19. By cloudlet 20, the execution time increased to 300.696 and by cloudlet 27 it rose to 301.19. Finally, the times revolved around 301 till the end of the simulation for cloudlet 39.

We also presented below the start and finish times for the simulations of scenarios i.e. for time-shared and the space-shared allocation policy. The following figures showed remarkable differences and variations. On the time-shared simulations (Figure 10 and Figure 11), the start time remained at 0.1 for both the first and second cloudlets for the simulation. The finish time started out at 80.270 and increased to 120.189 at cloudlet 16. Subsequently, it rose to 121.024 (cloudlet 22) and revolved around 121 till cloudlet 36. At cloudlet 37, it became 122.659.

For space-shared simulations (Figure 12 and Figure 13), the start time was 0.1 till cloudlet 15, then, from cloudlet 16, it became 100.481, and revolved around 100 till cloudlet 32, when it rose to 200, and maintained this value till cloudlet 19. On the other hand, for the first 20 cloudlets, the finish time started out at 100 and rose to 300 for cloudlet 16. This value was maintained till cloudlet 19. For the second 20 cloudlets, the finish time started out at 200 for cloudlet 20 and by cloudlet 32, it rose to 300, and maintained this value till the rest of the simulation.

Using this study, we sought to provide more understanding and make more sense of the two allocation policies of virtual machines. On time-sharing, there is the division of executing power (for CPU, logical processor, GPU) by different users in terms of operating system processes, threads and other requests or several existent pro-

Figure 10: First 20 Cloudlets for Time-shared Simulations

Figure 11: Second 20 Cloudlets for Time-shared Simulations

Figure 12: First 20 Cloudlets for Space-shared Simulations

Figure 13: Second 20 Cloudlets for Space-shared Simulations

cesses typically take turns in accessing the CPU. While space-sharing implies the division of memory space amongst several users. For the latter, goals are achieved by applying reusability i.e. storage space meant for input data are used again and do not necessary need to allocate more space during execution. Summarily, as Souza, et al. [2] puts it, "assign specific CPU cores to specific VMs (a space-shared policy); dynamically distribute the capacity of a core among VMs (time-shared policy)".

Conclusion and Future Directions

In this study, we attempted the implementation of some cloud scenarios using the CloudSim toolkit. Other software necessary for the installation and successful use of CloudSim, were also downloaded. Using two-dimensional graphs we plotted execution time versus 40 cloudlets used for all the scenarios. From the simulation, it is clear that increase in the length of instructions and input file size increases execution time. But that was for time-shared allocation policy. Interestingly, the execution time was almost halved when the allocation policy was changed to space-shared. In future, we would attempt to include other parameters that investigate client behaviours in both static and mobile contexts.

Bibliography

1. M Armbrust., *et al*. "Above the clouds: A Berkeley view of cloud computing". Univ California, Berkeley, Tech Rep UCB (2009): 07-013.
2. P M H Souza., *et al*. "Evaluating the Influence of the Client Behavior in Cloud Computing". *PLoS ONE* 11.7 (2016).
3. R N Calheiros., *et al*. "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms". *Software—Practice and Experience* 41 (2011): 23-50.
4. R Buyya., *et al*. "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities". *IEEE* (2009): 1-11.
5. F Fakhfakh., *et al*. "Simulation Tools for Cloud Computing: A Survey and Comparative Study". ICIS Wuhan, China (2017): 221-226.
6. W Tian., *et al*. "Open-source simulators for Cloud computing: Comparative study and challenging issues". *Simulation Modelling Practice and Theory* 58 (2015): 1-16.
7. A Beloglazov., *et al*. "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing". *Future Generation Computer Systems* 28 (2012): 755-768.
8. T Guérout., *et al*. "Energy-aware simulation with DVFS". *Simulation Modelling Practice and Theory* 39 (2013): 76-91.
9. D Goutam., *et al*. "The performance evaluation of proactive fault tolerant scheme over cloud using CloudSim simulator". 5th International Conference on the Applications of Digital Information and Web Technologies ICADIWT 5 (2014): 171-176.
10. F Fittkau., *et al*. "Cloud user-centric enhancements of the simulator CloudSim to improve cloud deployment option analysis". *Lecture Notes in Computer Science* (2012).

Assets from publication with us

- Prompt Acknowledgement after receiving the article
- Thorough Double blinded peer review
- Rapid Publication
- Issue of Publication Certificate
- High visibility of your Published work

Website: www.actascientific.com/
Submit Article: www.actascientific.com/submission.php
Email us: editor@actascientific.com
Contact us: +91 9182824667