engineering proceedings

MDPI

Proceeding Paper

Opportunistic Load Balancing for Virtual Machines Scheduling in a Cloud Environment †

Younes Khair [1,*] and Haouari Benlabbes [2]

[1] Department of Computer Science, Faculty of Exact Sciences, University of Tahri Mohammed, Bechar 08000, Algeria
[2] Department of Exact Sciences, Higher Normal School of Bechar, Bechar 08000, Algeria
* Correspondence: ynss.khair@univ-bechar.dz
† Presented at the 2nd International Conference on Computational Engineering and Intelligent Systems, Online, 18–20 November 2022.

Abstract: One of the Internet's fastest-growing technologies is cloud computing, which needs each user to share various resources and virtual computers to access the services they demand. The difficulty is in allocating the user's tasks among the many resources at hand while maintaining a balanced workload. In this work, we present the design of a new scheduling approach for cloud virtual machines in order to process events (related to resource overuse or underuse). This reduces the amount of effort consumed while maintaining the required performance levels in the cloud data center. To evaluate the effectiveness of the approach with real load paths, we conducted an experiment using the open-source cloud computing platform OpenNebula.

Keywords: cloud computing; virtual machine scheduling; opportunistic load balancing

Citation: Khair, Y.; Benlabbes, H. Opportunistic Load Balancing for Virtual Machines Scheduling in a Cloud Environment. *Eng. Proc.* **2023**, *29*, 1. https://doi.org/10.3390/engproc2023029001

Academic Editors: Abdelmadjid Recioui, Hamid Bentarzi and Fatma Zohra Dekhandji

Published: 10 January 2023

1. Introduction

Cloud computing has emerged in recent years as a new area of information technology. According to the National Institute of Standards and Technology, cloud computing is a model that allows users to access on demand and from anywhere on the Internet, to a shared pool of computing resources that can be provisioned quickly and flexibly, with minimal administrative effort and interaction with the service provider [1].

Virtualization technology is used in data centers to enable the existence of several virtual machines on the same physical machine. This capacity to personalize allows the service provider to maximize the investment of available physical resources, resulting in greater profits [2].

Cloud resource management activities involve examining the resources accessible in the cloud environment, offering access to end users, monitoring security, managing resource reservations, and monitoring all cloud layers. Because resource allocation management has a significant impact on the performance and efficiency of the cloud, as well as the cost of its users, it has become a difficult challenge to manage the various cloud resources, particularly to ensure cloud computing performance through strategic planning techniques that maintain cloud performance. This performance may be accomplished by maintaining a balanced distribution of workloads across cloud resources. One of the most challenging difficulties in cloud computing is finding a suitable level of task distribution among the many resources used in the cloud, namely: (one or more servers, hard drives, network links, or other resources). Thus, cloud service providers provide a means for distributing application requests to any number of data center-based application propagations. A balanced distribution aims to enhance processing speed while decreasing reaction time [3–5].

2. Related Work

The significant expansion of cloud computing's technology offerings is a reflection of both the system's user base and the tasks those users are requiring it to perform. As a result, a significant number of scheduling algorithms were developed and used in different cloud computing environments. The performance of these algorithms was then assessed based on a variety of different criteria [6].

Researchers have recently shown a great interest in scheduling in cloud computing. Several scheduling methods operating inside the cloud computing system have been suggested by various researchers.

For the purpose of scheduling distinct tasks, several heuristic algorithms have been developed and put to use in the cloud environment. The First Come First Serve basis algorithm, Min-Max algorithm, Min-Min algorithm, and Suffrage algorithm are among the most significant developments in heuristic algorithms. Other important developments include Greedy Scheduling, Shortest Task First, Sequence Scheduling, Balance Scheduling (BS), Opportunistic Load Balancing, and Min-Min Opportunistic Load Balancing [2–5,7].

In this work, we focus on using an algorithm that, for scheduling based on load balancing, offers the best performance in the cloud computing system. For a set of virtual machines, the algorithm was evaluated in two different scenarios. We looked into the subsequent performance scenarios: (1) the effect of load balancing throughput gain; (2) the effect of resource utilization rate.

3. OpenNebula

It is an open-source platform with an Apache 2 license that aims to develop a common cloud computing solution for controlling large and distributed infrastructures. Additionally, it has more capabilities, more adaptable methods, and enhanced cloud formation interoperability. OpenNebula offers hybrid clouds, which offer extremely flexible hosting options while integrating on-premises and public cloud architecture [6].

OpenNebula is currently on version (6.3.9). Its design is flexible, allowing interaction with different types of storage, network infrastructure, and hypervisor technologies. Figure 1 shows a diagram with the components grouped into three layers: (i) Drivers, (ii) Core, and (iii) Tools. These elements connect to each other through a set of APIs that provide system and cloud user interfaces [6].

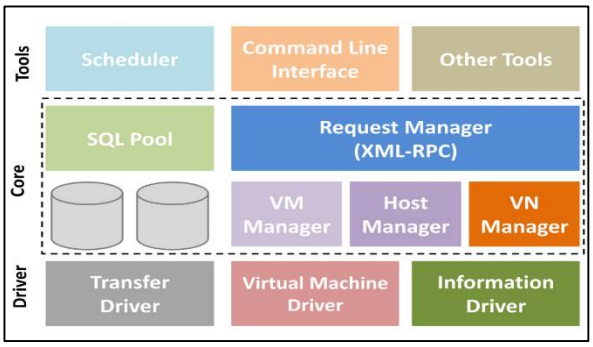

Figure 1. OpenNebula Architecture [8].

4. Opportunistic Load Balancing for Scheduling Virtual Machines in the Cloud Environment

The objective of this algorithm is to assign the work to the cloud host machine that is under the least increase in load when compared to other cloud hosts. The aim is to first identify each host machine's load distribution before allocating virtual machines to it. Then, the machine with the lower load level is chosen to perform the required function or task [9].

In our architecture concept (Figure 2), the opportunistic load balancing OLB is in charge of scheduling when these activities will be performed out on the pre-activated virtual machines. Hosts have two distinct resources (CPU and RAM). Each request for

allocation consists of two values that describe the virtual machine's execution requirements for CPU and RAM on one of the Cloud hosts. Additionally, OLB communicates with OpenNebula, using the OpenNebula Cloud API (OCA) which encapsulates access to OpenNebula's XML-RPC API and implicitly associated functions as shown in the definition in Section 3 [6]. To establish the connection, the OpenNebula XML-RPC address access methods are chosen and assigned. This integration allows OpenNebula to run OBL queries.

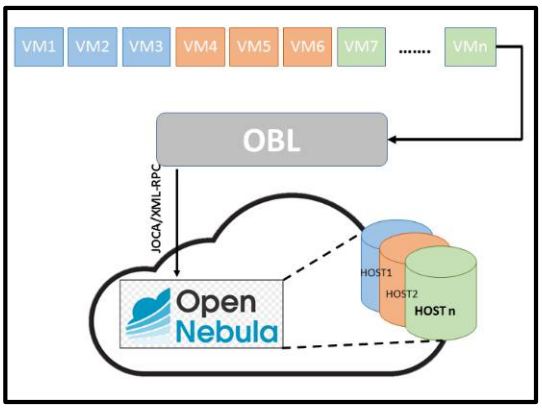

Figure 2. The architecture concept.

4.1. Scheduling Based on Opportunistic Load Balancing

Suppose the fact that there are $|N|$ similar resource host machines at the beginning, where H = {1, 2, . . . , H} The capacity of each host machine is *Ci* where $i \in$ H. Suppose that the capacities are given integer values, where each integer roughly corresponds to the CPU and RAM requirements of the host machine. Consider three machines, each hosting a distinct load from the others, H1 of C1 = 10%, H2 of C2 = 50%, and H3 of C3 = 90%, as illustrated in Figure 3.

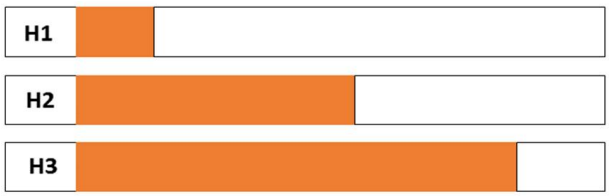

Figure 3. An initial situation for load balancing.

The capacity is proved by:

$$Ci = \frac{CPUuse * RAMuse}{100}$$

Consider that there are $|N|$ virtual machines, V = {1, 2, . . . , V}, each of which has a weight of W = {W1, W2, . . . Wn} and an end time for the activity that is being executed on it of T = {T1, T2, . . . Tn}, The virtual machine's weight is a representation of the funding necessary for its implementation. Whenever it uses a percentage of the host's resources, the weight is defined as follows:

$$Wi = \frac{CPUreq * RAMreq}{100}$$

To maximize the value of all virtual machines and ensure that the overall weight of virtual machines in each resource does not exceed that resource's capacity, the challenge of allocating virtual machines to resources occurs. Let u_{ij} The value reflects how significant this virtual machine is for the resource.

$$\sum_{i=1}^{H}\sum_{j=1}^{V} u_{ij}T_{ij}$$

4.2. The Proposed Scenario

Through the scenario (without and with the use of opportunistic load balancing), this algorithm was examined regarding the cost factors and rate of resource utilization. The parameter values for the scenario's results are summarized in Table 1.

Table 1. Specifications of the environment used for the experiments.

Host machines	Number of hosts	3
	RAM	2 GB
	Storage	250 GB
Virtual Machines	Number of Virtual Machines	10
	RAM	256 MB
	Number of processors	1

During the evaluation of the proposed planning algorithm, the characteristics of all the created virtual machines were installed and harmonized in the homogenous environment.

4.2.1. Load Balancing Throughput Gain

This is the average success rate for allocations on host machines. For each virtual machine that is allocated, the allocation rate of incoming tasks is calculated, and data is supplied at the end of each phase of allocation. The higher this factor, the better the algorithm performs. The value of the throughput gain is given as follows:

$$G_t = \sum_{alloc^i} (Exe_{time})$$

4.2.2. Resource Use Rate

The ratio of the host machine's total occupation to the totality of all the virtual machines' execution times is referred to as the resource utilization rate (in our scenario, it is the time taken to complete the execution).

$$Usage\ rate\ of\ each\ host\ machine = \frac{\begin{array}{c}The\ final\ time\ for\ the\ completion\\ of\ work\ on\ the\ host\ machine\end{array}}{Total\ runtime\ for\ all\ virtual\ machines} * 100$$

5. Results and Discussion

5.1. Load Balancing Throughput Gain

The behavior of throughput as a function of virtual machines in the what-if scenarios is shown in Figure 4. The results show that when a single resource is used, performance degrades as the number of allowances increase as load balancing is not enabled (there is no OBL). In this case, the host machines are not taken into account in terms of virtual machine allocation. In the opportunistic load-balancing scenario, average productivity returns to maximum levels despite an increase in the number of virtual machines.

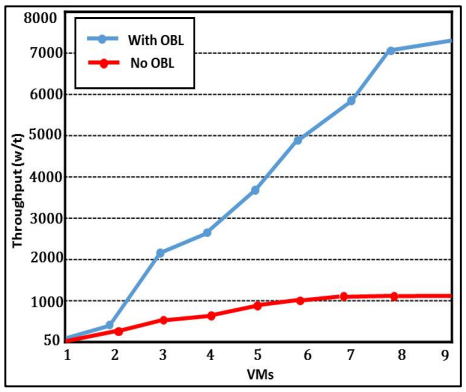

Figure 4. The behavior of throughput as a function of virtual machines.

5.2. Resource Use Rate

As we previously mentioned, the rate of resource utilization refers to the correlation between the overall host machine occupation and the execution timings of all virtual machines. Figure 5 shows that opportunistic load balancing has the best resource utilization rate.

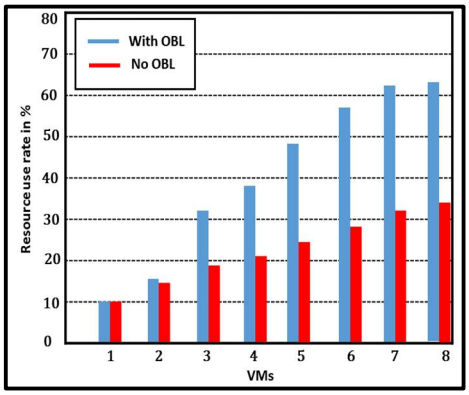

Figure 5. Resource use rate.

6. Conclusions

Through this research, we have planned to establish an algorithm that, for scheduling based on load balancing, offers the best performance in the cloud computing system. For a set of virtual machines, the algorithm was evaluated in two different scenarios. We looked into the subsequent performance scenarios: (1) the effect of load balancing throughput gain; (2) the effect of resource utilization rate.

Author Contributions: Conceptualization, H.B. and Y.K.; methodology, Y.K.; software, Y.K.; validation, Y.K and H.B.; formal analysis, Y.K.; investigation, Y.K.; resources, Y.K.; data curation, Y.K.; writing—original draft preparation, Y.K.; writing—review and editing, Y.K.; visualization, Y.K.; supervision, H.B.; project administration, H.B.; funding acquisition, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. William, V.; Broberg, J.; Buyya, R. Introduction to cloud computing. *Cloud Comput. Princ. Paradig.* **2011**, 1–44. [CrossRef]
2. Qi, Z.; Cheng, L.; Boutaba, R. Cloud computing: State-of-the-art and research challenges. *J. Internet Serv. Appl.* **2010**, *1*, 7–18.
3. Pietri, I.; Chronis, Y.; Ioannidis, Y. Multi-objective optimization of scheduling dataflows on heterogeneous cloud resources. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 361–368.
4. Sofia, A.S.; Kumar, P.G. Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II. *J. Netw. Syst. Manag.* **2018**, *26*, 463–485. [CrossRef]
5. Elaziz, M.A.; Xiong, S.; Jayasena, K.; Li, L. Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. *Knowl.-Based Syst.* **2019**, *169*, 39–52. [CrossRef]
6. Khair, Y.; Dennai, A.; Elmir, Y. Dynamic and elastic monitoring of VMs in cloud environment. *J. Supercomput.* **2022**, *78*, 19114–19137. [CrossRef]
7. Deafallah, A. A metaheuristic framework for dynamic virtual machine allocation with optimized task scheduling in cloud data centers. *IEEE Access* **2021**, *9*, 74218–74233.
8. Younes, K.; Dennai, A.; Elmir, Y. An Experimental Performance Evaluation of OpenNebula and Eucalyptus Cloud Platform Solutions. In *International Conference on Artificial Intelligence in Renewable Energetic Systems*; Springer: Cham, Germany, 2021.
9. Minxian, X.; Tian, W.; Buyya, R. A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e4123.