MMSIA: Improved Max-Min Scheduling Algorithm for Load Balancing on Cloud Computing

Tran Cong Hung Phan Thanh Hy Le Ngoc Hieu Nguyen Xuan Phi

Posts and Telecommunications Institute of Technology, Ho Chi Minh, Vietnam

conghung@ptithcm.edu.vn; phanthanhhy2793@gmail.com; occbuu@gmail.com; nguyenxuanphi@gmail.com

ABSTRACT

Cloud computing is one of the most advanced technologies in information technology, a convergence of many achievements in research and development and application of new technologies. Cloud computing has also helped to reduce the cost of small and medium enterprises based on cloud provider services. As cloud computing evolves rapidly, researching optimizations such as task execution time, completion time, responce time, and virtual machine resources (VMs) are tremendous challenges. This article proposes an MMSIA algorithm to improve the Max-Min scheduling algorithm, which improves the completion time of the requests by using the "learned learning" machine learning, by clustering size of requests and clustering utilization percent of VMs. The algorithm then assigns the largest cluster requests to the VM with the least utilization percent, which is repeated when the request list is empty. In particular, the MMSIA algorithm has improved the completion time. The simulation results show that the proposed MMSIA algorithm has improved the completion time compared to the three algorithms: Max-Min, Min-Min and Roud Robin.

Keywords

Max – Min, load balancing, improve Max – Min algorithm, comple, processing time.

1. INTRODUCTION

Cloud computing [1] is a broad research field that will bring great value to the costs of businesses around the world. Cloud computing [2] is a service model of information technology, inheriting the world's previous networks and distributed computing concepts that help users access resources data, complex data processing of systems such as google, facebook,…quickly, easily [3], [4]. Cloud computing is based on virtualization technology [5], through network services to provide users with the basic resources, application platform, software and other services. Cloud computing is changing the IT industry, changing the way we use software and hardware [6]. Make use of on-demand computing resources such as bandwidth, storage or software applications and computing is available. It is a new computational model, developed after computer distribution technology, grid computing, network storage, cluster technology and parallel computing [7]. Cloud computing is a new and evolutionary model most notable in computing [8]. Overload prediction in load balancing [9] has recently emerged as a promising solution, which monitors the congestion status of each road and distributes the data stream directly to the road less congestion is a significant issue. Load balancing [10] can be divided into 2 categories:

- Local load balancing is used to balance load forecasts in a central location.
- Global load is the management and control of requests from the customer automatically to the server through multiple data centers.

Because of the tremendous benefits of cloud computing, this article proposes an improvement to the Max - Min algorithm to reduce the response time and processing time of incoming requests in data centers cloud. The article is organized into 4 sections: Part 1: Introduction to cloud computing and load balancing. Part 2: Related works. Part 3: Propose algorithm: improved Max-Min algorithm. Part 4: Simulation results and evaluation. Part 5: Conclusion

2. RELATED WORK

The Round-Robin algorithm [11] is one of the oldest, simplest, most widely used and most widely used scheduling algorithms designed specifically for time-sharing systems. Round-Robin load balancing algorithms try to distribute loads to virtual machines in alternate rotation order. Actually, there are some modified versions of it that ALBA algorithm [12] is proposed for more robust load balancing through consideration of dynamic parameters. Algorithms (ALBA) are becoming popular because they are capable of changing their behavior based on current information, such as real-time server statistics while the algorithm is active.

The Min - Min algorithm [13] minimizes work completion time in each node, but the algorithm does not yet consider the workload of each resource. The traditional Min-Min algorithm is the basis for the current scheduling improvement algorithm (LBIMM) in cloud computing and overcome the disadvantages of the traditional Mix-Mix algorithm. After providing an improved algorithm for considering the workload of each resource, the algorithm improved (LBIMM) for better test results, shown in Table 1.

Table 1: Comparison of the results of the Min-Min and LBIMM algorithms. [13]

	Min – Min	**LBIMM**
Number of Task	10	10
Number of Response	5	5

Makespan (sec)	19.625	12.5
Average Resource Utilization Rate	45.29%	82.29%

The Max-Min algorithm [14] selects the task with the maximum expected completion time and assigns that task to the resource with a minimum overall execution time. The goal of the algorithm is to minimize system vulnerabilities, to meet the requirements for lifecycle, to limit load balancing between virtual machines and to predict server response times.

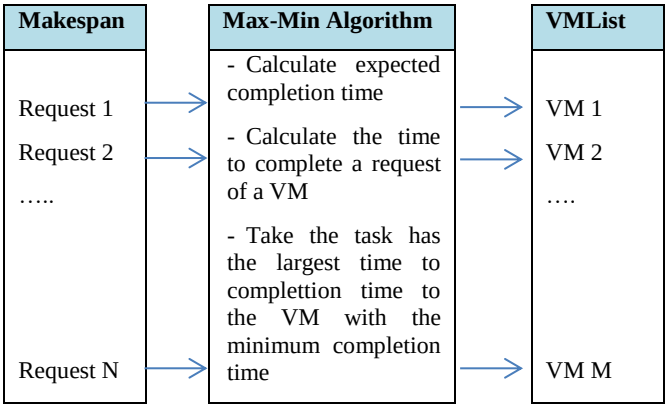

Figure 1: Max-Min scheduling algorithm model

Max-Min scheduling algorithm schema:

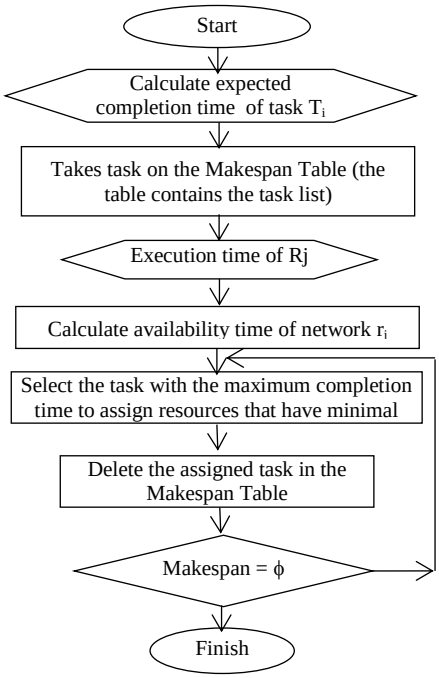

Figure 2. Max-Min scheduling algorithm

3. PROPOSED ALGORITHM: Max-Min Scheduling Improved Algorithm (MMSIA)

3.1 Assumptions

- Load balancers know the requests are placed in the request list (Makespan) table.
- The load balancer also knows the list of virtual machines, the percentage used by the Virtual Machine.
- The MMSIA algorithm assumes 10 requirements and 5 virtual RAM, memory, and different CPUs.

3.2 Target of MMSIA algorithm

- MMSIA algorithm will reduce the processing time required compared to the existing Max-Min and Max-Min scheduling algorithms.
- Reduce the processing time of all incoming requests.
- Speed up the processing of virtual machine requests without load imbalance.

3.3 MMSIA algorithm model

3.3.1 Model of MMSIA

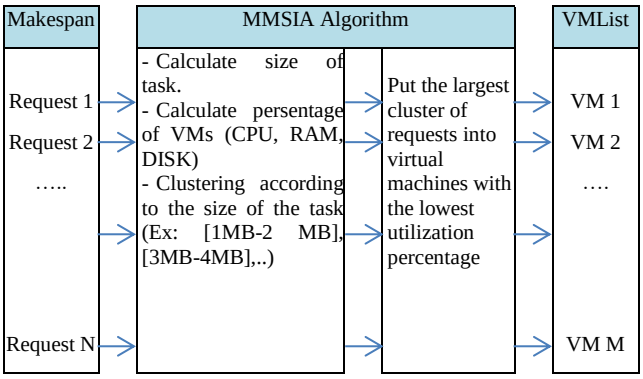

Figure 3. MMSIA algorithm based on supervised learning

Description of the MMSIA algorithm:

- The idea is to put a "supervised learning" algorithm in machine learning into the Max-Min scheduling algorithm for clustering requests according to size and computing the percentage of virtual machines in different clusters from which Max-Min scheduling algorithm performs the following two functions:

  - The comparison function and assignment requests the largest file sizes for virtual machines with the least amount of space used.
  - After the assignment is complete, then request processing and show the result and recalculate the current utilization percentage of the VMs.

- The MMSIA algorithm will repeat until the Makespan tables are empty. At that time the requirements will be processed faster, shorten the finishing time, increased load balancing capability for cloud computing.

3.3.2 Diagram of MMSIA Algorithm

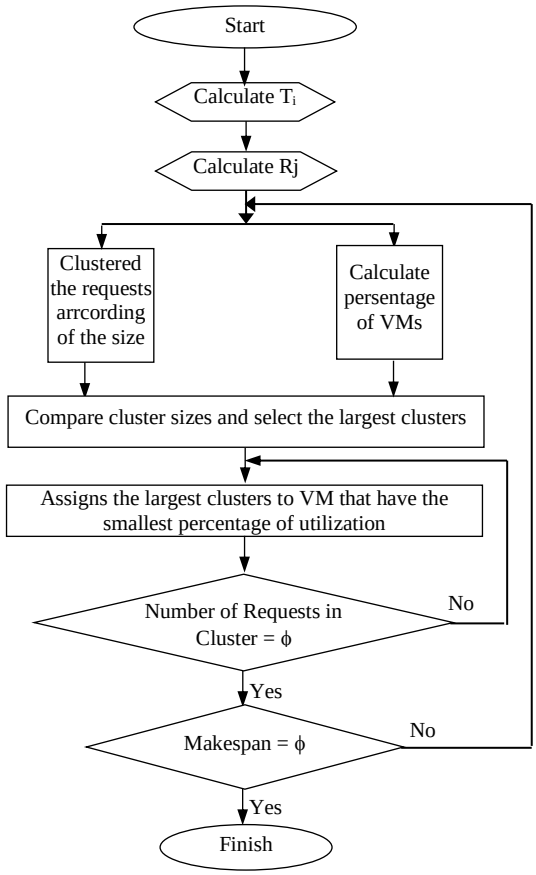

Figure 4. MMSIA algorithm

4. SIMULATION RESULTS AND EVALUATION

4.1 Simulation Environment

This section describes how to implement the proposed algorithmic simulation of load balancing based on the size of the requests from which clusters and assign the appropriate virtual machines to perform. MMSIA is a new way of solving the load balancing problem, namely the improvement of the Max-Min scheduling algorithm to improve the processing time, maintain the life of the nodes in the cloud system, thereby out how to coordinate resources more effectively.

Emulator the cloud environment using the CloudSim library and programming in the JAVA language. The cloud emulator environment is between 3 and 10 virtual machines, and creates a random request environment for services on the cloud. Includes virtual cloud services, cloudSim provisioning and user provisioning services for testing. Installing the Max-Min scheduling algorithm, and the MMSIA algorithm on the simulation environment, and testing the results. The parameters as well as the script given are based on the request process of the browser in the cloud environment.

Parameters of the simulation model:

Table 2. Datacenter configuration parameters

Datacenter	Host in Datacenter
- Number of host in datacenter: 5 - Not used SAN Storage - Architecture (arch): x86 - Operating System (OS): Linux - Processing (VMM): Xen - TimeZone: +7 GMT - Cost of processing: 3.0 - Cost of memory usage: 0.05 - Cost of using capacity: 0.1 - Cost of bandwidth usage: 0.1	Each host in the datacenter is configured as follows: - The CPU has 4 cores, each core has a processing speed of 1000 (mips). - Ram: 2048 (MB) - Storage: 1000000 - Bandwidth: 10000

- Virtual machines have the same configuration when initialized:

Table 3. VMs configuration parameters

Capacity of Storage (MB)	RAM (MB)	MIPS	Bandwith	Number of CPU	VMM
10000	512	250	1000	1	Xen

- The requests (WebRequest) are represented by Cloudlet in CloudSim and the size of Cloudlets is randomly generated using the JAVA random function. Cloudlet numbers are 25, 50, 100, 1000, respectively.

Table 4. Requests configuration parameters

Length	File Size	Output Size	Number of CPU (PEs)
3000 ~ 1700	5000 ~ 45000	450 ~ 750	1

The proposed algorithm is constructed using the Cloudlet value aggregation algorithm for evaluation. The algorithm is as follows:

```
public double getAverageSize()
{
    double res=0;
    res = this.getCloudletLength() * 0.5 + this.getCloudletOutputSize()*0.2
    + this.getCloudletFileSize()*0.3;
    return res;
}
```

Use the algorithm to calculate the average virtual machine percent:

```
public double getAverageRequestedResources()
{
    double res =0;
    double per_mips=(this.getCurrentRequestedTotalMips() / this.getMips())
    * 100 ;
    double per_ram = (this.getCurrentRequestedRam() / this.getRam() ) *
    100;
    double per_storage = (this.getCurrentAllocatedSize()/this.getSize()) *
    100;
    res = (per_mips + per_ram + per_storage)/3;
    return res;
  }
```

4.2 Results

Experimental simulations will be performed in about 500 requests back, with 4 times and each time there will be 5 virtual machines and the number of requests is 25, 50, 100 and 500 respectively. This experiment will run on 4 different algorithms, namely Max-Min scheduling algorithm, RoundRobin, Min-Min algorithm, and MMSIA algorithm.

Table 5. Experimental simulation results

Number of Requests	Max-Min	Round Robin	Min- Min	MMSIA
25	210.59	276.82	338.44	185.74
50	175.8	476.69	1380.55	165.63
100	2268.1	597.29	623.59	537.79
500	2664.22	3374.22	3743.97	2625.95

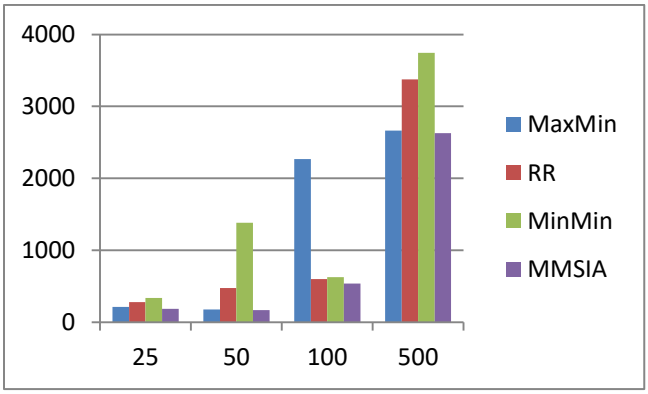

Figure 5. The graph compares the completion time of the algorithms

Figure 5 shows that the MMSIA algorithm has the lowest completion time of all virtual machines compared to the three algorithms: Max-Min, Min-Min, Roud Robin. This simulated experiment is only done on a group of virtual machines, not to mention expanding the set of virtual machines to reduce the load in case of necessity, assuming that the group of VMs handles as many requests as possible, if it exceeds new expansion. However, this simulation is only done in small models and the number of requests is low. By aggregating clusters of requests by file size, VMs are faster to process, and the system categorizes requests, which in turn provides the lowest possible percentage of virtual machines to process.

5. CONCLUSIONS

The MMSIA algorithm operates based on clustering size of request and utilization percent of VMs, from which to allocate large request for virtual machines with the smallest percentage of utilization (CPU, Ram, and Disk) based on Max-Min mechanism.

This paper demonstrates that the MMSIA algorithm has improved the completion time of requests. The results of the MMSIA algorithm are better than the Max-Min algorithm (Figure 5). In addition, algorithm results are significantly less than algorithms: Min-Min, Roud Robin.

As such, the MMSIA algorithm has improved cloud performance, specifically improving completion time in the cloud. Enables faster access to cloud services, in other words, to improve the quality of service for users.

REFERENCES

[1] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu and G. Xu,(Feb. 1 2016), *"A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment,"* in IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 2, pp. 305-316.

[2] GIBET TANI, H. and C. EL AMRANI, *"Smarter Round Robin Scheduling Algorithm for Cloud Computing and Big Data*", Journal of Data Mining and Digital Humanities, 2018. Special Issue on Scientific and Technological Strategic Intelligence (2016).

[3] Y. F. Wen and C. L. Chang,(2014), *"Load balancing job assignment for cluster-based cloud computing,"* Sixth International Conference on Ubiquitous and Future Networks (ICUFN), Shanghai, 2014, pp. 199-204.

[4] Bui Thanh Khiet, Nguyen Thi Nguyet Que, Ho Dac Hung, Pham Tran Vu, Tran Cong Hung, "*A Fair VM Allocation for Cloud Computing based on Game Theory*", Proceedings of the 10th National Conference on Fundamental and Applied Information Technology Research (FAIR'10), Da Nang, Viet Nam, 17-18/08/2017.

[5] J. Zhang, Q. Liu and J. Chen, *"An Advanced Load Balancing Strategy for Cloud Environment,"* 2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Guangzhou, 2016, pp. 240-243.

[6] Matthias Sommer, Michael Klink, Sven Tomforde, Jörg Hähner, "*Predictive Load Balancing in Cloud Computing Environments Based on Ensemble Forecasting*" 2016 IEEE International Conference on Autonomic Computing (ICAC2016),Wurzburg, Germany, 17-22 July 2016, pp. 300 - 307.

[7] G. Shao and J. Chen, "*A Load Balancing Strategy Based on Data Correlation in Cloud Computing*," 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC), Shanghai, 2016, pp. 364-368.

[8] Ashok Kumar Upadhaya, C.K. Jha, Shikha Pandey, "*Suboptimal Mechanism For Load Balancing In CloudEnvironment*", 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET), Chennai, India, 16-18 Feb. 2017.

[9] P. Wang, H. Xu, Z. Niu, D. Han and Y. Xiong, *"Expeditus: Congestion-Aware Load Balancing in Clos Data Center Networks*," in IEEE/ACM Transactions on Networking, vol.25, no. 5, pp. 3175-3188, Oct. 2017.

[10] C. Jayashri, P. Abitha, S. Subburaj, S. Y. Devi, Suthir S and Janakiraman S, *"Big data transfers through dynamic and load balanced flow on cloud networks*," 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2017, pp. 342-346.

[11] Vo Van Khang,(2014), *"Research on Load Balancing in Cloud Computing"*, Hong Bang International University, Ho Chi Minh City, Viet Nam.

[12] Furqan Alam, Vijey Thayananthan, Iyad Katib, *"Analysis of Round-robin Load-balancing Algorithm with Adaptive and Predictive Approaches*", in 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, UK, 31 Aug.-2 Sept. 2016.

[13] Tran Cong Hung, Nguyen Xuan Phi, "*STUDY THE EFFECT OF PARAMETERS TO LOAD BALANCING IN CLOUD COMPUTING*", in International Journal of Computer Networks & Communications (IJCNC) Vol.8, No.3, May 2016.

[14] Rajwinder Kaur,Pawan Luthra, *"Load Balancing in Cloud System using Max Min and MinMin Algorithm*", IJCA Proceedings on National Conference on Emerging Trends in Computer Technology(IJCA Journal), 2014.

MMSIA: Improved Max-Min Scheduling Algorithm for Load Balancing on Cloud Computing

First Author	**Name**	**Tran Cong Hung**
	Title* (Prof., Assoc. Prof., Asst. Prof., Dr., Mr. Ms. etc.)	Assoc. Prof
	Research Interests	His main research areas are B – ISDN performance parameters and measuring methods, QoS in high speed networks, MPLS. He is, currently, Associate Professor PhD. of Faculty of Information Technology II, Posts and Telecoms Institute of Technology in HOCHIMINH, Vietnam
	Homepage URL	http://tranconghung.com/
Second Author	**Name**	**Phan Thanh Hy**
	Title* (Prof., Assoc. Prof., Asst. Prof., Dr., Mr. Ms. etc.)	Master's degree.
	Research Interests	His main research areas are load balancing on cloud computing
	Homepage URL	
Third Author	**Name**	**Le Ngoc Hieu**
	Title* (Prof., Assoc. Prof., Asst. Prof., Dr., Mr. Ms. etc.)	Master's degree
	Research Interests	His main research areas are load balancing on cloud computing
	Homepage URL	
Fourth Author	**Name**	**Nguyen Xuan Phi**
	Title* (Prof., Assoc. Prof., Asst. Prof., Dr., Mr. Ms. etc.)	PhD Sutudent
	Research Interests	His main research areas are: load balancing on cloud computing, optimizing the performance of cloud computing, networking.
	Homepage URL	