



Application of Heuristic Algorithm in Accomplishing Computer Instructions

*¹Maipan-uku J.Y., ¹Limanko I. H., ²Mishra A., ¹Abdulganiyyu A., ¹Kawu A. A., ¹Elijah J. and ¹Awal A.

¹Department of Computer Science, Ibrahim Badamasi Babangida UniversityLapai, Nigeria

²Department of Computer Science, Baze University, Abuja, Nigeria

ABSTRACT

Focusing on the Infrastructure as a Service (IaaS), scheduling seems to be the most primary problem of cloud computing as tasks (cloudlets) scheduling have to be carry out simultaneously and efficiently in order to meet the cloud consumers' expectations. To realize this goal, tasks have to be executed as fast as possible in order to reduce the total completion time (makespan) and ensuring efficient resource (virtual machine) utilization. Many algorithms had implemented to remedy this problem. Algorithms such as Max-min is implemented which finds the tasks with maximum execution time and allocates to the resource with it minimum completion time. The limitation of Max-min scheduling algorithm is of high make span and low resource utilization when the number of tasks with maximum execution time is more than the number of tasks with minimum execution time. To address this problem, a new algorithm called Max-Range is proposing which allocates tasks based on range of completion time. A tool kit known as cloudsims (cloud simulator) was used to implement and compare the performances of the proposed algorithm with existing algorithms. The results of the experiment indicate that the proposed algorithm is able to produce good quality solutions by reducing the make span and reduce the cost of execution as compared to the standard Max-min scheduling algorithm.

Keywords: Cloud computing, Max-min Scheduling, Task Scheduling.

INTRODUCTION

Cloud computing is a means of accessing resources over the internet without owning the infrastructure. Cloud computing has gained a great significance and popularity in the IT sector due to majority of organization, companies and individuals being unable to buy the resources(hardware and software) because of the high cost involved in getting them. Cloud computing services accessed according to the Service Level Agreement (SLA). The Service Level Agreement is the contract between the Cloud Service Providers (CSP) and the Cloud Service Consumers (CSC). The main purpose of cloud computing is to help the cloud consumers (users) manage resources such as hardware, software, network and data resources, this responsibilities are shifted to the Cloud Service Providers (CSP). CSPs includes; Windows Azure, Amazon web services and Google compute engine. Service models are the reference models on which cloud computing based. Service models are the various ways by

which Cloud service providers (CSP) offer services such as: Software as a service (SaaS), Platform as a service (PaaS) and Infrastructure as a service (IaaS). All of the above services have their different level of user flexibility and control, which allows users to run applications and store data online..

IaaS model is the most basic level of the service model, it provides access to fundamental resources such as virtual machine, physical machines, servers, virtual storage (host). PAAS model provides the runtime environment for applications, development and deployment tools and web servers (build) while SAAS model allows users to run existing online application(consume).This research focus on IaaS.

Received 13July, 2022

Accepted 04 November, 2022

Address Correspondence to:

jymaipanuku@ibbu.edu.ng

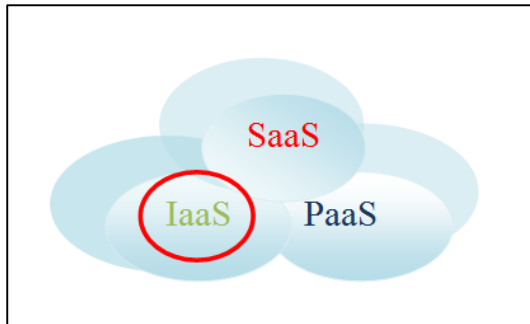


Fig 1. Cloud computing services

Task scheduling is the major challenge faced in cloud computing (as per IaaS) as the number of cloud consumers are increasing due to high difference in operating system, architecture of the system, cloud service providers, and cloud service consumers. Task scheduling refers to the method in which tasks are been allocated to computing devices (VMs) for execution. The best scheduling algorithm is that which manages availability of resources such as processor, CPU and storage memory as well as utilizes the resources to its maximum and gives the minimum finish time (makespan). Max-min task scheduling algorithm is one of the effective tasks scheduling algorithm that considers the tasks with maximum execution time and assigns it to resources with the minimum finish time (makespan). The max-min repeats the steps until all tasks are engaged. The main drawback of the max-min algorithm is its biased approach towards selecting the tasks with maximum execution time and assigning them to the resources with minimum finish time. Using this approach, the tasks with minimum execution time will have to wait for the larger tasks to finish executing and leaving some resources idle. The proposed algorithm (max-range) aimed at addressing the limitations of the Max-min algorithm. The proposed algorithm will also be compare with existing algorithms.

Different task scheduling algorithms are in existence today. The various scheduling algorithms in cloud computing environment are dubious and this leads to a continuous slowdown in the speed of executing tasks. The past years have witnessed an exciting movement for cloud tasks scheduling and resources allocation. It has maintained its growth in the computer networks, telecommunication and the storage market. This improved its concentration with the introduction of innovative and groundbreaking scheduling algorithms, by various researches (Min-MinKeshket *al.*, 2014;Koket *al.*,2016; Max-

MinElzeki., 2012;B &Manjaiah, 2007, Navdeep &Khushdeep, 2015;Konjaanget *al.*, 2017;Hung *et al.*, 2019; Min-Max, Zhou & Hu, 2014; Game TheoreticDuan *et al.*, 2014;Wei *et al.*, 2010; FCFS Aditya Marphatia, 2013) and other algorithms have made resource allocation and task scheduling in the cloud more efficient. This includes an improved league championship algorithm in (Abdulhamid *et al.*, 2014) and Symbiotic Organism Search optimization based task scheduling in (Abdullahi *et al.*, 2016). Though cloud computing has gone through significant reforms and transformations, however, the rapid growth in information and the increasing demand for cloud services in recent times calls for a proactive for provision of an optimal scheduling techniques to provide better and cost-effective cloud services to meet the increasing demands and requirements of cloud consumers/users. However, these researchers have contributed immensely in making Max-Min algorithm more effective in terms of makespan optimization. Thus, our proposed approach aims at addressing it by introducing a new dimension to make Max-Min more effective and proactive in selection and scheduling both tasks with minimum and maximum execution time simultaneously without given any priority.

Proposed Algorithm

Satisfying consumer's expectation on cloud computing require simultaneous execution of cloudlets. Ideologically, we might acquire greatest schedule if we are capable of allocating cloudlets to the VMs in which can be administered in minimum likely processing time. Max-Min algorithm method, as discussed in earlier do this, but allocating cloudlets in order of Max-Min resolve in large value of makespan, high computational cost, and low resource (VMs) utilization, if longer cloudlets are more than the smaller cloudlets. This is the major drawback of Max-Min.

However, in this research work, we present our proposed algorithm (Max-Range) in Table 1. Firstly, all the cloudletss arranged in increasing order. This means, cloudlets of minimum completion time are in the front of the queue and cloudlets with maximum completion time at the end of the queue. Secondly, range of completion time for executing the entire jobs on the available VMs is compute as a pivot, then, VM is chosen for execution.

Table 1: Max-Range Pseudo Code

1. While there are cloudlets in CloudletList
2. for all submitted cloudlets in the set; c_i
3. for all VMs; vm_j
 - o Calculate completion time (CT_{ij}) = $et_{ij} + rt_j$; (for each cloudlet in all VMs)
 - o arrange all completion time in ascending order
 - o find the range of cloudlets completion time
 - o for all cloudlets with completion time < the range
 - assign the c_i to the vm_j that produce it minimum completion time for execution
 - o for all cloudlets with completion time > the range
 - assign the c_i to the vm_j that produce it minimum completion time for execution
4. Update the CloudletList
5. Update ready time (rt_j) of the selected vm_j
6. Update ct_{ij} for all c_i
7. End While

METHODOLOGY

Due to the difficulties in implementing, testing, and evaluating the efficiency of the proposed algorithm with the existing ones using a testbed, CloudSim toolkit, a simulation environment for cloud computing was use to experiment and ascertain the performance of these algorithms. Some scientific workflows such as Inspiral and Montage were used as dataset. The simulation was run on Intel® Pentium (R) CPU, 500GB HDD and 4GB RAM on 64-bit operating system (Windows 10 Home). Table 2. Shows the cloudsims configuration adopted.

Table 2. Cloudsim Configurations

Entities		Values
Workflows	Inspiral	50, 100, 1000
	Montage	50, 100, 1000
Data center		1
Virtual Machines		15
CPU/VM		1

Performance Metric

Makespan as one of the performance metrics is used to major the efficacy of the proposed Extended Min-Min algorithm with the standard Min-Min, Max-Min, Round Robin and FCFS.

Makespan

Makespan is defined as the time when surfaces the newest task. This parameter shows the quality of assignment of resources from the executional time perspectives. It can be calculated as shown in the Equations below;

If $C_i = c_1, c_2, c_3, \dots, c_m$, is the set cloudlet submitted to the scheduler, and $VM_j = vm_1, vm_2, vm_3, \dots, vm_n$, be the set of virtual machines available at the time of cloudlet arrival,

$$Makespan = \max \{ completion [j] \mid j \text{ in Resource} \} \tag{1}$$

$$Completion Time (CT_{ij}) = (ET_{ij}) + (R_j) \tag{2}$$

Where ET_{ij} is the expected execution time of the cloudlet c_i on machine vm_j and r_j is the ready time of vm_j . i.e. the time when vm_j becomes ready to execute c_i .

Execution Cost

Since cloud computing adopt pay as you use mechanism, then the total execution cost for a list of cloudlets is of paramount for evaluating system performance. Eqn 3.2 represents mathematical formation of this parameter.

$$TC = \left(P_j * PC + \left(\sum_{f \in Fin_j} Size(f) + \sum_{f \in Fout_j} Size(f) \right) * TrC \right) \tag{Eqn. 3.2}$$

Where TC is the total cost, P_j is the processing time of the job j , PC is the processing cost, and TrC is the cost of transferring the input files (Fin_j) and the output files ($Fout_j$).

Table 3. Cost of Virtual Machines.

Item	Cost(\$)
Cost of processing	5.0
Cost of BW	0.1
Cost per memory	0.05
Cost per memory	0.1

RESULTS AND DISCUSSION

Makesplan

Tables 4 and Fig. 2 depict these parameters in LIGO Inspirial dataset. LIGO stands for Laser Interferometer Gravitational wave observatory, which is a dataset used in LIGO Scientific Collaboration.

The value of this parameter reflects the finish time for all submitted jobs. It is noticeable that the Max-Range algorithm bar had the lowest values among the other algorithms in all datasets. This is because Max-Range reduces the time needed to transfer the data by locating the best cloud provider, which leads to faster jobs execution and gives lowest makespan.

Table 4. Ligo Inspirial with 10VMs

Cloudlets	Max-Min	Min-Min	Max-Range
50	8991.75	11904.23	7650.41
100	15616.64	71126.32	6750.49
1000	47217.78	55748.81	39249.01

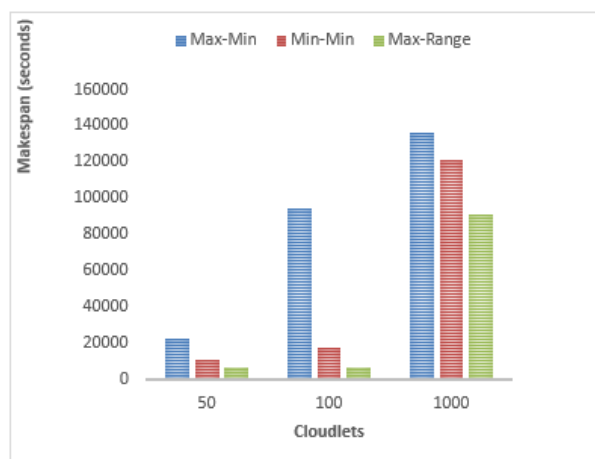


Fig 2: Makespan based on Ligo-Inspirial

Table 5 and Figures 4 depict these parameters in Montage dataset. It shows a set of experiments that done on three groups of jobs using the three algorithms. The value of this parameter reflects the finish time for all submitted jobs. It is noticeable that the Max-Range algorithm bar had the lowest values among the other algorithms in all datasets. This is because Max-Range reduces the time needed to transfer the data by locating the best cloud provider, which leads to faster jobs execution and gives lowest makespan.

Table 5 Montage with 10VMs

Cloudlets	Max-Min	Min-Min	Max-Range
50	530.73	483.08	271.46
100	897.7	1803.69	433.33
1000	4424.16	4888.72	4135.58

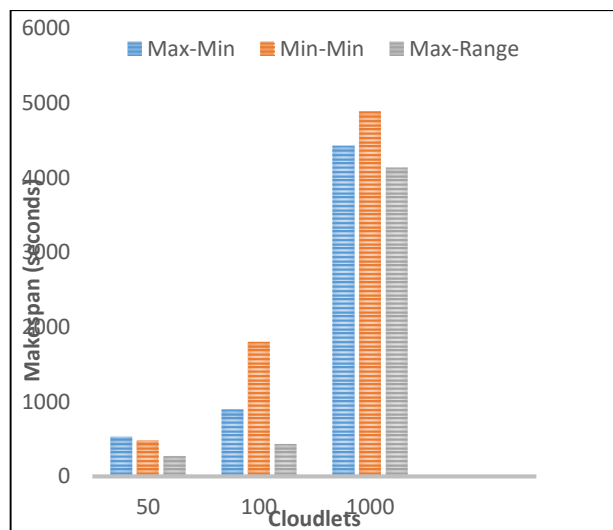


Fig 4. Makespan based on Montage

Execution Cost

The cost is the total execution cost for getting a task done by a virtual machine, since cloud computing charge depending on how the user consume services. The cost of execution is important as it allows the users' job execution in lesser cost.

Tables 6 and 7 depict these parameters in LIGO Inspirial dataset and Montage dataset respectively. It shows a set of experiments conducted on three groups of jobs using the three algorithms. The value of this parameter reflects the cost of execution for the three algorithms for all submitted jobs. It is noticeable that the Max-Range algorithm had the lowest values among the other algorithms in all datasets. This is because Max-Range reduces the cost needed to execute tasks on virtual machines.

Table 6. Costs on Ligo Inspirial with 10VMs

Cloudlets	Max-Min	Min-Min	Max-Range
50	58996.21	58996.81	58994.53
100	105480.50	105479.34	105480.28
1000	1142174.18	1142168.00	1142174.83

Table 7. Costs on Ligo Inspiral with 10VMs

Cloudlets	Max-Min	Min-Min	Max-Range
50	2647.13	2646.80	2646.54
100	5590.69	5590.73	5589.03
1000	58790.43	58804.49	58790.65

Conclusion

Efficient scheduling can yield better cloud services. The desire of each cloud user is to access cloud services within the shortest time by reducing the finish time (makespan) and the cloud service providers desire to maximize resource utilization. Achieving such goals is difficult due to the dynamic nature of the cloud environment. To overcome the challenge in scheduling tasks in cloud environment, there is a need for an efficient scheduling algorithm.

This research proposed an algorithm (Max-Range). This algorithm (Max-Range) is compared to existing algorithms such as Min-Min and Max-Min. We simulated the proposed approach using Cloudsim toolkit, which is java-based simulator. The simulator was used to execute different number of jobs by using two existing scheduling algorithms (Max-Min and Min-Min); the results of the experiment were compared to the result of the proposed algorithm. The outcome of the experiment showed that the proposed (Max-Range) algorithm was able to produce less makespan, less execution cost and efficient resources utilization as compared to the two algorithms (Max-min and Min-min).

Recommendations

This research focused only on the makespan and Cost Effectiveness in cloud computing environment. Future study can carry out the experiment using different dataset such as Cybershake, Sipt, Heft, Epigenomics, Floodplain and also consider other parameters such as resource utilization, load balance and also compare the proposed algorithm to other algorithms such as FCFS, Round Robin, Improved Max-min, OLB e.t.c.

REFERENCES

Abdulhamid, S. M., Latiff, M. S. A. & Idris, I. (2014). Tasks scheduling technique using League Championship Algorithm for makespan minimization in IaaS cloud. *ARPN Journal of Engineering and Applied Sciences*, 9(12).

Abdullahi, M., Ngadi, M. A. & Abdulhamid, S. M. (2016). Symbiotic Organism Search optimization based task scheduling in cloud computing environment. *Future Generation Computer Systems*, 56. <https://doi.org/10.1016/j.future.2015.08.006>

Aditya Marphatia, A. M. (2013). Optimization of FCFS Based Resource Provisioning Algorithm for Cloud Computing. *IOSR Journal of Computer Engineering*, 10(5). <https://doi.org/10.9790/0661-1050105>

B, S. & Manjaiah, D. H. (2007). An Improved Task Scheduling Algorithm based on Max-min for Cloud Computing. *International Journal of Innovative Research in Computer and Communication Engineering (An ISO Certified Organization)*, 32972(2).

Duan, R., Prodan, R. & Li, X. (2014). Multi-objective game theoretic scheduling of bag-of-tasks workflows on hybrid clouds. *IEEE Transactions on Cloud Computing*, 2(1). <https://doi.org/10.1109/TCC.2014.2303077>

Elzeki, O. M., Rashad, M. Z. & Elsoud, M. A. (2012). Overview of Scheduling Tasks in Distributed Computing Systems. *International Journal of Soft Computing and Engineering*, 2(3).

Hung, T. C., Hy, P. T., Hieu, L. N. & Phi, N. X. (2019). MMSIA: Improved max-min scheduling algorithm for load balancing on cloud computing. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3310986.3311017>

keshk, A. E., El-Sisi, A. B. & Tawfeek, M. A. (2014). Cloud task scheduling for load balancing based on intelligent strategy. *International Journal of Intelligent Systems & Applications*, 6(5), 25–36.

Kok, J., Maipan-uku, J. Y. & Kennedy, K. (2016). An Efficient Max-Min Resource Allocator and Task Scheduling Algorithm in Cloud Computing Environment. *International Journal of Computer Applications*, 142(8). <https://doi.org/10.5120/ijca2016909884>

Konjaang, J. K., Ayob, F. H. & Muhammed, A. (2017). An optimized max-min scheduling algorithm in cloud computing. *Journal of Theoretical and Applied Information Technology*, 95(9).

Navdeep, K. & Khushdeep, K. (2015). Improved max-min scheduling algorithm. *Journal of Computer Engineering*, 17(3), 42–49.

Wei, G., Vasilakos, A. v., Zheng, Y. & Xiong, N. (2010). A game-theoretic method of fair resource allocation for cloud computing services. *Journal of Supercomputing*, 54(2). <https://doi.org/10.1007/s11227-009-0318-1>

Zhou, Z. & Hu, Z. (2014). Task scheduling algorithm based on greedy strategy in cloud computing. *Open Cybernetics and Systemics Journal*, 8(1). <https://doi.org/10.2174/18741110x01408010111>