

Performance Evaluation of Min-Min and Max-Min Algorithms for Job Scheduling in Federated Cloud

R.Jemina Priyadarsini

Research Scholar,
St.Joseph's College, Tiruchirappalli-620002

L.Arockiam

Associate Professor(CS),
St.Joseph's College, Tiruchirappalli-620002

ABSTRACT

Cloud computing is a large-scale, economic driven, distributed computing paradigm where abstracted, virtualized, managed computing power, scalable, storage, platforms and services are delivered on demand to customers over internet. It refers to applications delivered as services through the internet and hardware and systems software at data centres providing such services. As Cloud computing is a major platform, it is important to understand its implications on customers' applications or Cloud deployed systems. Hence, simulation tools are critical not only to evaluate Cloud's performance but also to further develop Cloud computing. This study investigates the performance of min-min and min-max scheduling algorithms using CloudSim software.

General Terms

Cloud Computing

Keywords

Cloud Computing, CloudSim, Min-min, Max-min, Virtualization and Scheduling

1. INTRODUCTION

Cloud computing, coined in 2007, is now a hot topic as it offers flexible, dynamic, IT infrastructures, computing environments guaranteed by QoS and configurable software services [1]. Cloud computing includes researchers and engineers from different backgrounds like database, Grid computing and software engineering. Cloud Computing is envisioned as next-generation IT enterprise architecture, due to its advantages in IT history: on-demand self-service, network access, rapid resource elasticity, usage-based pricing, location independent resource pooling, and risk transference [2]. Cloud Computing is transforming how businesses use information technology as it is a disruptive technology with profound implications.

Cloud Computing uses Internet-based technologies to provide services [3], starting from the cloud as an internet metaphor based on computer network diagrams to the abstract complex infrastructure it covers. It is also a commercial evolution of academic-oriented grid computing. Generally cloud computing is offered with a usage-based model where you pay for cloud resources alone that a specific computation needs. Computations needing additional resources request them from the cloud (up to cloud's overall capacity) [4]. Due to on-demand resource provisioning and utility based pricing, service providers are bound to increase resource utilization and reduce operating costs. Virtualization abstracts away physical hardware details providing virtualized resources for high-level applications [5]. A virtualized server is called a Virtual Machine (VM). Virtualization is the basis of cloud computing having the ability to pool computing resources from server clusters and dynamically assign or reassign virtual resources to applications when needed. Virtualization benefits include improved use, managing, and mainframe systems

reliance. Users with varied OS requirements can easily share a virtualized server as OS upgrades are staged across VMs to minimize downtime while guest software failures are isolated to the VMs it occur in.

Virtualization improves system security and reliability through isolation of multiple software stacks in their VMs [6]. Security is improved as intrusions are confined to VMs, while reliability is enhanced as software failures in a VM do not affect others. Virtualization technologies include various mechanisms and techniques to decouple architecture and user-perceived hardware behavior and software resources from physical implementation [7].

Two performance issues are raised by virtualization: its cost and performance benefits it offers. Recent hardware and software development were partially motivated to reduce virtualization cost. Popular microprocessors including Intel, AMD, and IBM Power have hardware features supporting virtualization to reduce performance cost [8]. Virtualization technology ensures security services with better protection by placing them in protected VMs [9]. Task scheduling is a famous combinatorial optimization problem. The goal is in determining a sequence where tasks are executed while conforming to constraints. Implementations are labelled as centralized or decentralized, static or dynamic, or hybrid, all through their own strengths and limitations. Usually, dynamic load-balancing mechanisms ensure improved performance than static ones, but they encounter higher overhead due to systems information having to be updated on-the-fly. Task scheduling is unaffected in a static setup as it is based on a-priori knowledge.

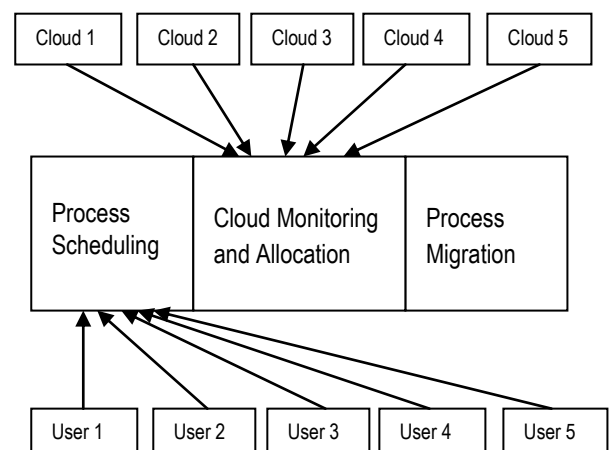


Fig. 1 Process Scheduling

To begin with, scheduling algorithms were implemented in grids due to grids lower performance; it is now scheduled in a cloud [11] as shown in Figure 1. Application scalability is the main advantage of moving to clouds. Unlike grids, cloud resources scalability allows real-time resources provisioning

to meet application requirements. Traditional cloud computing scheduling used direct tasks of users as an overhead application base.

Some common scheduling algorithms are [12]:

- First Come First Serve Algorithm: First Job in queue is served. This is a simple and fast algorithm. .
- Round Robin algorithm: Here scheduling processes are dispatched in a FIFO manner but given limited CPU time called time-slice or quantum.
- Min–Min algorithm: This chooses small tasks for execution first, which in turn delays bigger tasks for long.
- Max – Min algorithm: This chooses large tasks for execution first leading to delays in smaller tasks.
- Most fit task scheduling algorithm: Here fit best tasks in the queue are executed first. However, it also has a big failure rate.
- Priority scheduling algorithm: Here, each process is assigned a priority, which is allowed to run. Equal-Priority processes are scheduled in FCFS order. Shortest-Job-First (SJF) algorithm is a special general priority scheduling algorithm.

Job priority is important in scheduling as some jobs must be serviced earlier than others as they cannot be in the system for long. A job scheduling algorithm considers job priority. To offset this, some researchers considered jobs priority scheduling algorithm [13]. In cloud environments various attributes are to be considered. It means a specific job scheduling algorithm in cloud environments should be attentive to a job’s multi-attribute and multi-criteria properties. There are many Multi-Criteria Decision-Making (MCDM) and Multi-Attribute Decision-Making (MADM) modules based on mathematical modeling.

CloudSim simulation layer supports modeling and simulation of virtualized Cloud-based data center environments including VMs dedicated management interfaces, storage, memory and bandwidth. This layer handles issues like hosts provisioning to VMs, managing application execution, and monitoring dynamic system state. A cloud provider wants to study the efficiency of different policies for allocating its hosts to VMs needs to implement the strategies at this layer [14]. By extending this layer’s entities, a Cloud application developer performs the following: (i) generates a mix of workload request distributions and application configurations; (ii) models cloud availability scenarios performing robust tests based on custom configurations; and (iii) implements custom application provisioning techniques for clouds and their federation.

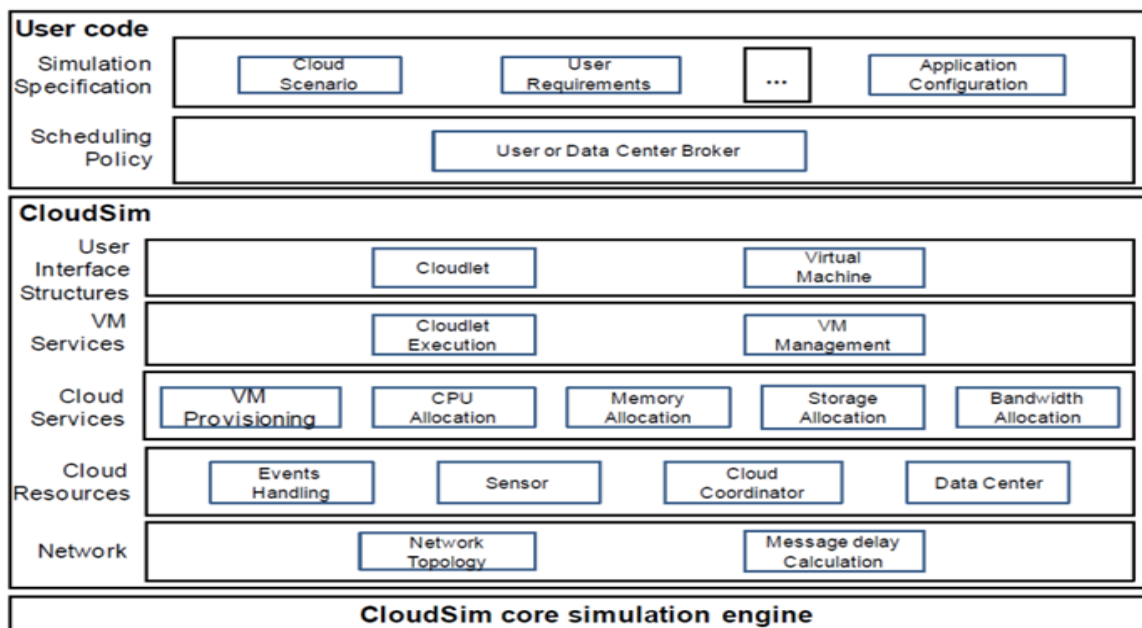


Fig. 2 Layered CloudSim Architecture

Figure 2 shows a multi-layered design structure of the CloudSim architecture [15]. First layer is that the User Code which makes entities such as applications, number of users and their applications, virtual machines and others available to hosts. Second layer is the CloudSim which can be made up of many fundamental classes (java). These are some key components required for the implementation of the CloudSim such as:

- Cloudlet: The complexity of cloud based applications can be measured in terms of their computational requirements by CloudSim. Cloudlet class models these application services where the cloudlet scheduler is used to implement policies for determining the processing

power share among many Cloudlets present in a virtual machine.

- Datacenter: This class modelled the cloud based hardware or infrastructure services. Datacenter Characteristics class is used to provide information regarding the data centre resources.
- Datacenter Broker: This models a broker who is responsible for mediating negotiations between Cloud Service Provider (CSP) and the customer. These can be driven by QoS requirements. The broker studies various CSP and also negotiates the allocation of various resources and services as per QoS requirements in order to provide best one for the customer.

- iv. Host: This class is used to model physical resources like assigning the policy to share processing power, memory and bandwidth among various virtual machines.
- v. Network Topology: This class provides information to induce the network behavior during the simulation.
- vi. Cloud Coordinator and Sensor: This is responsible for periodic monitoring of the internal state of data centre resources. Dynamic load - shredding decisions are undertaken for which it uses the dynamic performance information. To monitor the performance, parameters implementation of sensor is necessary.
- vii. Virtual Machine (VM): This class models a VM in the cloud which includes allocation of all VM on the hosts and so it is selected based on the requirements for a VM deployment. This class is responsible for allocation of processor cores among the VMs [16].

CloudSim has novel features including: (i) supporting modeling and simulation of huge scale cloud computing infrastructure, including data centers on one physical computing node; and (ii) a self-contained platform to model service brokers, data centers, scheduling, and allocations policies. CloudSim's unique features include: (i) availability of virtualization engine, which creates and manages multiple, independent, and co-hosted virtualized services on data center nodes; and (ii) adaptability to switch between time-shared and space-shared processing cores allocation to virtualized services [17]. These CloudSim features quicken development of new algorithms, methods and protocols in cloud computing thereby contributing to quicker paradigm evolution.

Computing is being transformed as a model consisting of services, commoditized and delivered similar to traditional utilities like water, electricity, gas and telephony. This study discusses cloud computing and scheduling schemes. Cloudsim software simulates performance of min-min and max-min algorithms. The study is organised as follows. Section 2 discusses related works in literature. Section 3 explains methodologies used. Section 4 discusses experiments and results and Section 5 concludes the paper.

2. RELATED WORKS

A scrutiny of energy consumption in cloud computing was presented by Baliga et al [18]. It considered public and private clouds and included energy consumption due to switching and transmission as also data processing and storage. Cloud computing enables better energy-efficient use of computing power, specially when computing tasks are of low intensity or infrequent. Though, cloud computing can consume more energy than conventional computing under some circumstances where each user performs all computing on personal computers.

A comprehensive survey on latest developments in service-oriented network virtualization to support Cloud computing particularly from a network perspective and cloud convergence through NaaS was presented by Duan et al [19]. The SOA principle was specifically introduced, and reviewed recent research progress on utilizing SOA to support network virtualization in telecommunications and Internet. Then, a network-cloud convergence framework based on service-oriented network virtualization was presented and given a survey on key technologies to realize that NaaS, mainly focused on state of the art network service description, discovery and composition.

The achievement of cloud computing services aimed at scientific computing workloads was analyzed by Iosup et al [20]. An empirical performance evaluation of four commercial cloud computing services was performed including Amazon EC2, and compared through trace-based simulation, performance characteristics and cost models of clouds and other scientific computing platforms. The performance was evaluated for both general and MTC-based scientific computing workloads. The study concluded that the current clouds need an order of magnitude in performance improvement to be useful to scientific community. They also showed what improvements should be first considered to address the discrepancy between offer and demand.

A cloud computing empowered WPS framework for earth observation data management proposed by Chen et al [21] consisted of client and WPS layers, which also included a WPS server layer and a cloud computing layer. Cloud computing environment was open-source software Apache Hadoop based. The proposed cloud computing's three layers, enabled WPS and were outlined, followed by a workflow processing user's task using the three layers. Technological implementation details were explained. An experiment processing Moderate Resolution Imaging spectro radiometer (MODIS) data revealed that WPS could be enabled in a cloud computing environment.

A Hierarchical Attribute-Set-Based Encryption (HASBE) by extending ciphertext-policy Attribute-Set-Based Encryption (ASBE) with a hierarchical users structure to realize scalable, flexible, and fine-grained access control of outsourced data in cloud computing was proposed by Wan et al [22].

Both analytical and simulation modeling were used to address cloud computing systems complexity by Khzaei [23]. The analytical model included functional sub models, whose results were combined iteratively to obtain an accurate solution. The new models incorporated important features of cloud centers like batch arrival of user requests, resource virtualization and realistic servicing steps to obtain important performance metrics like task blocking probability and total waiting time incurred on user requests. Also, experiment revealed important insights towards capacity planning to control delay in servicing user's requests. Optimal networked cloud mapping problem as a mixed integer programming issue was formulated by Papagianni et al [24]. This indicated objectives related to the resource mapping procedure's cost efficiency, while going by user requests for QoS-aware virtual resources. This adopted a heuristic method to address the issue. The proposed approach's efficiency was illustrated in a simulation/emulation environment permitting a flexible, structured and comparative performance evaluation.

Green Computing Concept (GCC) schemes based on networks were surveyed by Xiong et al [25]. GCC's concept and history were introduced first, with the focus then on cloud computing's challenges and requirements. Cloud computing must become green, meaning provisioning of cloud services while considering energy consumption under energy consumption criteria called GCC. Further, recent work in networks based GCC including microprocessors, virtualisation technology, task scheduling algorithms, cooling systems, networks and disk storage was introduced. Then GCC works from a research group were presented at Georgia State University.

A theoretical energy-optimal mobile cloud computing framework under stochastic wireless channel was provided by Zhang et al [26]. The aim was conserving energy for mobile

devices by optimally executing mobile applications in it (mobile execution) or offloading it to the cloud (cloud execution). Numerical results show that much energy could be saved for mobile devices by optimally offloading mobile applications to the cloud. The proposed theoretical framework and numerical investigations would shed lights on mobile cloud computing system implementation under stochastic wireless channels.

A priority-based method to merge parallel workloads in the cloud was proposed by Liu et al [27]. Virtualization technologies partitioned each node's computing capacity into two tiers, foreground VM tier (with high CPU priority) and background VM tier (with low CPU priority). Experiments revealed that the proposed parallel scheduling algorithm outperformed common algorithms like extensible Argonne scheduling system in data center settings. The procedure was practical and effective to consolidate data centre's parallel workload.

A performance study of network I/O workloads in a virtualized cloud environment was presented by Mei et al [28]. Virtual Machine Monitor's (VMM) present implementation, fails to ensure sufficient performance isolation to guarantee resource sharing effectiveness across multiple VM instances running on one physical host machine, specially when applications on neighbouring VMs compete for computing and communication resources. The main characteristics for consideration in scheduling issues in hybrid clouds when scheduling workflows and a survey of some scheduling algorithms used was introduced by Bittencourt et al [29]. To assess the communication channels influence on job allocation, the impact of available bandwidth on performance of some scheduling algorithms was compared and evaluated.

A baseline, Kepler Graphics Processing Units (GPU) built General-Purpose (GPU) GPGPU cloud system to explore hardware potential while improving task performance was described by Hu et al [30] which elaborated a general scheme defining the cloud system to a cloud layer, server layer, and GPGPU layer. Hardware and task features, scheduling and execution mechanisms of every layer were illustrated. This provided a better understanding of general-purpose computing on a GPGPU cloud.

A new low-power task scheduling algorithm for large-scale cloud data centers was presented by Xu et al [31]. A winner tree made data nodes as leaf nodes and final winner reducing energy consumption was chosen. Large-scale cloud data center's complexity was considered and task comparison coefficient defined to ensure a more reasonable task scheduling strategy. Experiment and performance analysis revealed that the new algorithm improved node utilization and reduced overall cloud data center's power consumption.

A new cloud scheduler which considered user requirements and infrastructure properties was proposed by Abbadi and Ruan [32]. Virtual resources using physical resources matching requirements were hosted without involving users with cloud infrastructure details. As proof-of-concept, the prototype built on OpenStack was presented. The prototype implemented the new cloud scheduler.

Agarwal et al [33] presented max-flow min-cut based salient object detection in 3D point cloud that result from Structure from Motion (SfM) pipeline where the SfM pipeline generates noisy point cloud due to unwanted scenes captured along with the object present in image dataset. The background points

being sparse and not meaningful, it becomes necessary to remove them. Any further processes utilize the cleaned up model that have no hindrance from the noise removed. The algorithm was completely autonomous and does not need any user input. Proposed method was tested based on the Indian historical models reconstructed through SfM. The results were evaluated in terms of selectivity and specificity.

Mu et al [34] proposed an improved version of basic max-min ant system (MMAS). It uses the grey information which was obtained from pheromone matrix to build grey model, predict and control the maximum and minimum trail limits real-timely. Also it makes some other parameters of algorithm that were controlled by cloud association rules. Algorithm avoid effectively the slow convergence that were caused possibly by implementing max-min trail limit strategy and the early stagnation of search, appease contradiction between the convergent speed and searching scope dynamically. The simulation result for JSP shows the validity of it.

Chen et al [35] introduced an improved Load Balanced algorithm on the ground of Min-Min algorithm to reduce the Makespan and increase resource utilization (LBIMM). Cloud providers offer computer resources to users on a pay-per-use base. To accommodate the demands of different users, they may offer different levels of quality for services. Cost per resource unit depends on the services selected by the user. So the user receives guarantees regarding the provided resources. To observe promised guarantees, user-priority was considered the proposed PA-LBIMM so that user's demand gets satisfied more completely. Finally algorithm was simulated using Mat lab toolbox. The simulation results show that the improved algorithm leads to significant performance gain and achieve over 20% improvement on both VIP user satisfaction and resource utilization ratio.

3. MATERIALS AND METHODS

In this study, the performance of the min-min and max-min scheduling algorithm are investigated. Both the algorithms are widely used.

3.1 Min-Min

Minimum completion time for each task in min-min is computed for all machines. The task with overall minimum completion time is chosen and assigned to corresponding machine. The newly mapped task is removed and the process is repeated till all tasks are mapped [36].

Min-min is a simple and fast algorithm capable of good performance. Even GA "seeds" a population with a min-min chromosome to ensure good performance. Min-min schedules "best case" tasks first generating good schedules. Assigning small task first is its drawback. Thus, smaller tasks are executed first and then few larger tasks are executed while many machines are idling, resulting in poor machine use. Min-min establishes minimum completion time for unscheduled jobs (similar to MCT), and then assigns jobs with minimum completion time (hence min-min) to a processor which offers it.

The Min-Min algorithm [32]

for all tasks T_i in meta-task M_v

for all resource R_j

$C_{ij} = E_{ij} + r_j$

do until all tasks in M_v are mapped

for each task in M_v , find the earliest completion time and the resource that obtains it

find the task T_k with the minimum earliest completion time

assign task T_k to the resource R_l that gives the earliest completion time
delete task T_k from M_{V9}
update R_l 10.

update C_{ij} for all i
end do.

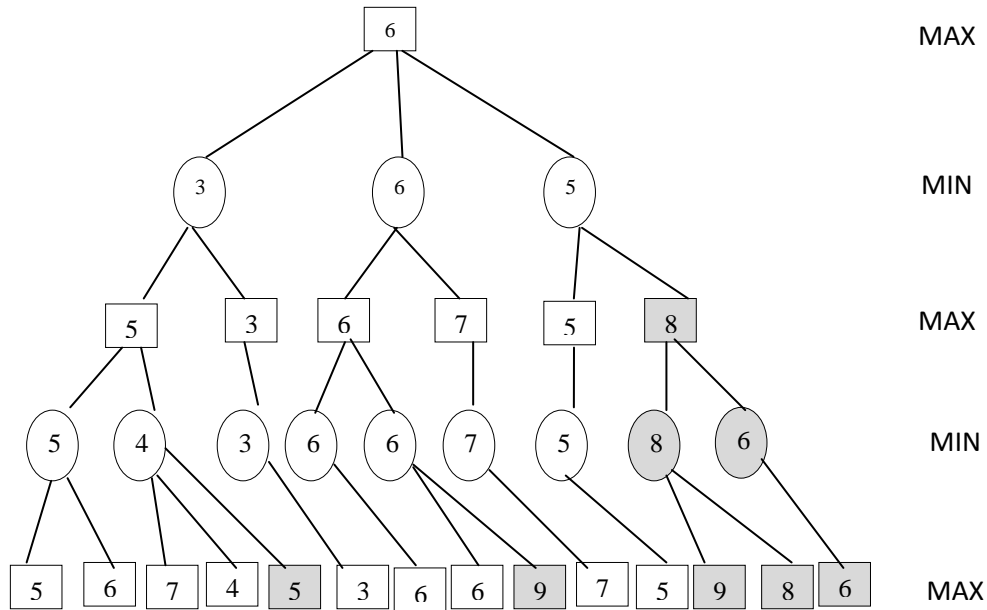


Fig. 3 Example Max-Min Tree

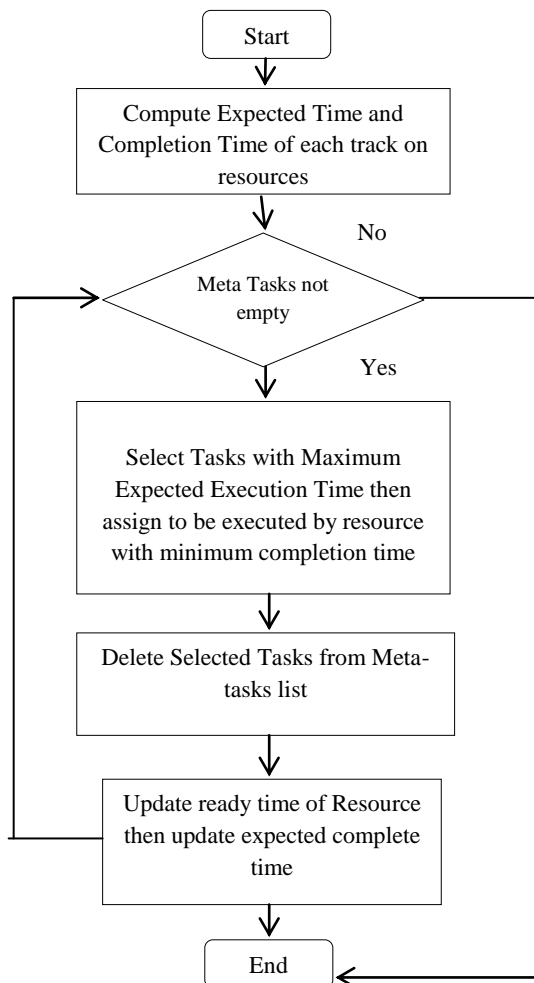


Fig. 4 Flowchart of the Max-Min Algorithm

3.2 Max-Min

The max-min heuristic is similar to min-min algorithm. The minimum completion times set is calculated for each task and that with overall maximum completion time is selected and assigned to a corresponding machine [36]. This algorithm does better than min-min algorithm where when short tasks outnumber long ones. For e.g. if there is one long task, max-min algorithm executes short tasks concurrently with long task. Max-min [38] is similar to min-min. Again each job's minimum completion time is established, but a job with maximum minimum completion time is assigned a corresponding processor. Figure 3 is an example tree for Max-Min algorithm.

Figure 4 shows the flowchart for Max-Min algorithm. The algorithm calculates each resource's submitted tasks expected completion time. Then task with overall maximum expected execution time (Largest Task) is assigned a resource with minimum overall completion time (Slowest Resource) [39].

The Max-Min algorithm :

for all submitted tasks T_i in meta-task
for all resource R_j
 $C_{ij} = E_{ij} + r_j$
find task T_k cost maximum execution time (largest task)
assign task T_k to the resource R_j which gives the minimum completion time (Slowest resource)
remove task T_k from Meta-task set
update r_j for selected R_j
update C_{ij} for all j
While meta-task not empty
Find task T_k cost maximum completion time
Assign task T_k to resource R_j which gives minimum execution time (Faster Resource)
Remove Task T_k from meta-task set
Update r_j for selected R_j
Update C_{ij} for all

4. RESULTS AND DISCUSSION

CloudSim software is used to run the min-min and max-min algorithms. Experiments are conducted with sixty tasks assigned to Cloud with 4 resources. The resources are located at two data centres. Each resource has 1 CPU with 1 Gb RAM. Each task is of size ranging from 1-7 units.

Table 1. Makespan

Method	Makespan
Min-min	47
Max-min	44

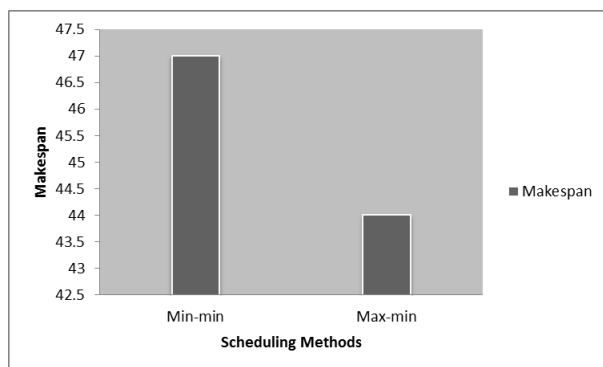


Figure 5 Makespan

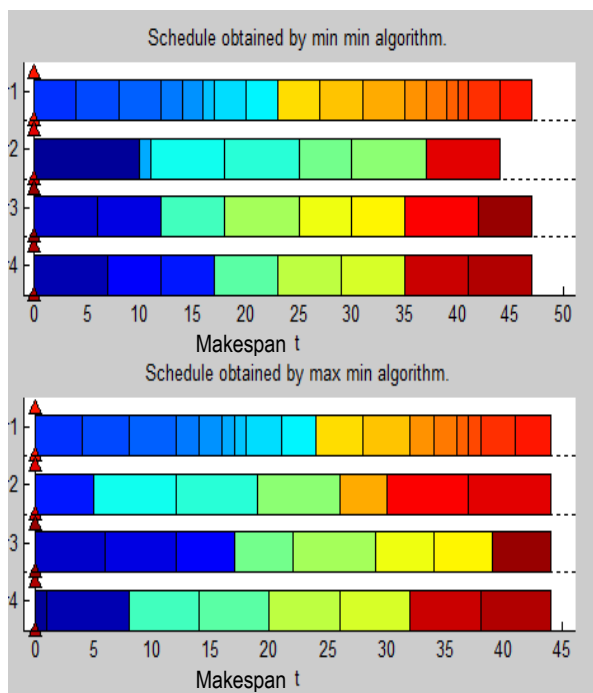


Figure 6 Makespan

It is observed from the results in Figure 5 and Figure 6, the max-min scheduling algorithm achieves lower makespan when compared to min-min scheduling. Thus task scheduling algorithm is for mapping jobs submitted to cloud environment onto available resources in such a way that the total response

time, the makespan, is minimized as shown in Table 1. [39].Efficient unitization of resources helps to achieve high performance with the help of optimization techniques like Particle swarm optimization (PSO) and Evolutionary algorithm (EA) [40].

5. CONCLUSION

Min-min and max-min algorithms are commonly applied in small distributed systems. When small tasks outnumber large tasks in a meta-task, max-min algorithm schedules tasks, where system makespan depends on how many small tasks are executed concurrently with large ones. Experiments conducted with sixty tasks are assigned to Cloud with 4 resources and the makespan calculated. It is concluded from the above experiments that the scheduler who works on various policies, is most important in a cloud and it is on the basis of the scheduler that VMs are allocated to specific hosts. And it is observed that for the small distributed system, max-min scheduling achieves better makespan.

6. REFERENCES

- [1] Lizhe Wang, Jie Tao, Marcel Kunze, Alvaro C. Castellanos, Kramer.D, & Karl, “Scientific cloud computing:Early definition and experience” 10th IEEE International Conference on High Performance Computing and Communications September 2008, pp. 825-830.
- [2] Wang C,Wang Q., Ren, K., and Lou. W, “Privacy-preserving public auditing for data storage security in cloud computing”, INFOCOM, 2010 Proceedings IEEE ,pp. 1-9, IEEE.
- [3] Marinos. A ,and Briscoe. G, “Community cloud computing”,Springer Berlin Heidelberg, pp. 472-484.
- [4] Grossman. R. L, “The case for cloud computing. IT professional”,2009, pp. 23-27.
- [5] Zhang. Q., Cheng, L., and Boutaba, R,“Cloud computing: state-of-the-art and research challenges”,Journal of Internet Services and Applications,1(1),2010,pp7-18.
- [6] Uhlig. R, Neiger. G, Rodgers. D, Santoni, A. L, Martins. F. C, Anderson. A. V and Smith. L, “ Intel virtualization technology Computer”, 2005,pp 48-56.
- [7] Figueiredo .R., Dinda. P.A, and Fortes. J, “Guest Editors' Introduction: Resource Virtualization Renaissance Computer”,2005,pp 28-31.
- [8] Mergen. M. F, Uhlig. V, Krieger, O, and Xenidis. J, “Virtualization for high-performance computing”. ACM SIGOPS Operating Systems Review, 40(2),pp 8-11.
- [9] Payne. B. D, Carbone. M, Sharif. M., and Lee. W. Lares, “An architecture for secure active monitoring using virtualization In Security and Privacy”, 2008 IEEE Symposium, pp. 233-247.
- [10] Dominique. A. Heger, “Optimized Resource Allocation & Task Scheduling Challenges in Cloud Computing Environments”,2010.
- [11] Sujit Tilak, and Patil.D, “A Survey of Various Scheduling Algorithms in Cloud Environment”, International Journal of Engineering Inventions, September 2012,pp.36-39
- [12] Salot.P, “A Survey Of Various Scheduling Algorithm In Cloud Computing Environment, IJRET, 2013.

- [13] Ghanbari. S, and Othman. M, “A Priority based Job Scheduling Algorithm in Cloud Computing”, *Procedia Engineering*, pp. 778-785.
- [14] Calheiros R. N, Ranjan R, Beloglazov .A, De Rose, C. A., and Buyya R, “ CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, *Software: Practice and Experience*, January 2011,pp. 23-50.
- [15] Howell F, and McNab R. “ SimJava: A discrete event simulation library for java. *Simulation Series*”, 30, 1998,pp 51-56.
- [16] Calheiros R. N, Ranjan R, Beloglazov .A, De Rose, C. A, and Buyya R, “ CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms”, *Software: Practice and Experience*, January 2011,pp. 23-50.
- [17] Calheiros R. N, Ranjan. R, De Rose C. A, and Buyya R , “Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services”,
- [18] Baliga J, Ayre R. W., Hinton, K., & Tucker R. S, “ Green cloud computing: Balancing energy in processing, storage, and transport”,*Proceedings of the IEEE*,2011,pp.149-167.
- [19] Duan Q, Yan Y, and Vasilakos A.V, “A Survey on Service-Oriented Network Virtualization Toward Convergence of Networking and Cloud Computing”, *IEEE Transactions on Network and Service Management*, 2012,pp. 373-392.
- [20] Iosup A , Ostermann S, Yigitbasi M. N, Prodan R , Fahringer T, and Epema D. H. (2011). “Performance analysis of cloud computing services for many-tasks scientific computing *Parallel and Distributed Systems*”, *IEEE Transactions on*, 2011,pp. 931-945.
- [21] Chen Z, Chen N, Yang C, and Di L, 2012, “Cloud Computing Enabled Web Processing Service for Earth Observation Data Processing”.
- [22] Wan Z, Liu J. E, and Deng R. H, “HASBE: A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing”, *IEEE Transactions on Information Forensics and Security*,2012, pp.743-754.
- [23] Khazaei H. 2012, “A fine-grained performance model of cloud computing centers”.
- [24] Papagianni C, Leivadreas A, Papavassiliou S, Maglaris V, and Monje A,2013, “On the optimal allocation of virtual resources in cloud computing networks”.
- [25] Zhang W., Wen Y., Guan K., Kilper D., Luo H., & Wu D, 2013, “Energy-Optimal Mobile Cloud Computing under Stochastic Wireless Channel”.
- [26] Xiong N., Han W., & Vandenberg A., “Green cloud computing schemes based on networks: a survey”, *Communications, IET*, 2012, pp.3294-3300
- [27] Liu X, Wang, C, Zhou, B, Chen, J, Yang, T, and Zomaya A, 2012, “Priority-Based Consolidation of Parallel Workloads in the Cloud”.
- [28] Mei Y, Liu L, Pu X, Sivathanu S, and Dong X, “ Performance analysis of network I/O workloads in virtualized data centers”,2013.
- [29] Bittencourt L. F, Madeira E. R and Da Fonseca N. L, (2012),“Scheduling in hybrid clouds”,*Communications*
- [30] Hu L, Che X, and Xie Z, “GPGPU cloud: A paradigm for general purpose computing”, *Tsinghua Science and Technology*, 2013,pp.22-23.
- [31] Xu X, Wu J, Yang G, and Wang R,“Low-power task scheduling algorithm for large-scale cloud data centers”, *Journal of Systems Engineering and Electronics*,2013,pp. 870-878.
- [32] Abbadi I, and Ruan A, “Towards Trustworthy Resource Scheduling in Clouds”, June 2013,pp.973-984.
- [33] Mu F, Yuan X. H, Wang C. G, and JING, Y. (2010). “Ant-colony-genetic algorithm with adaptive parameters based on grey prediction and normal cloud.*Control Theory and Applications*”, 6, 005.
- [34] Chen H, Wang F, Helian N, and Akanmu, G. “User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing”. In *Parallel Computing Technologies (PARCOMPTECH)*, February 2013 National Conference .pp. 1-8, IEEE.
- [35] Wu M.Y, Shu, W, and Zhang H. “Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems”. In *Heterogeneous Computing Workshop, 2000,HCW 2000, Proceedings. 9th* ,pp. 375-385, IEEE.
- [36] Parsa S, and Entezari-Maleki R., “RASA: A new task scheduling algorithm in grid environment”. *World Applied sciences journal*, 7, pp 152-160.
- [37] Miriam D. D. H, and Easwarakumar K. S. “A Double Min Min Algorithm for Task Metascheduler on Hypercubic P2P Grid Systems”. *International Journal of Computer Science Issues*,2010,Vol 7(5), pp 8-18.
- [38] Bhoi U, and Ramanuj P. N. “Enhanced Max-min Task Scheduling Algorithm in Cloud Computing”.
- [39] O. M. Elzeki, M. Z. Reshad, M. A. Elsouid, “Improved Max-Min Algorithm in Cloud Computing ”, *International Journal of Computer Applications (0975 – 8887) Volume 50 –No.12*, July 2012.
- [40] R.Jemina and L.Arockiam, “Failure Management in cloud: An Overview” *International Journal of Advanced Research in Computer and Communication Engineering*, vol 2, issue 10, October 2013, pp 4003 -4008.

7. BIOGRAPHY

JeminaPriyadarsini.R is doing her research in the Department of Computer Science, St.Joseph’s College Autonomous), Tiruchirappalli, Tamil Nadu, India. She has 12 years of teaching experience. She has attended many International and National Conferences, and also various Seminar and Workshops. Her broad area of research is ‘Cloud Computing’. She is presently working on Reliability Issues in Cloud Computing. Her area of interest is Software Engineering, Data Structures and Algorithms, Unified Modeling Language, Computer Graphics and Mobile Computing.

Dr. L. Arockiam is working as Associate Professor in the Department of Computer Science, St.Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India. He has 25 years of experience in teaching and 18 years of experience in research. He has published more than 194 research articles in the International & National Conferences and Journals. He has also presented 2 research articles in the Software Measurement European Forum in Rome. He has chaired many technical sessions and delivered invited talks in National and

International Conferences. He has authored 3 books. His research interests are: Cloud Computing, Big Data, Cognitive Aspects in Programming, Data Mining and Mobile Networks.

He has been awarded "Best Research Publications in Science" for 2009,2010 & 2011 and ASDF Global "Best Academic Researcher" Award from ASDF, Pondicherry for the academic year 2012-13 and also the "Best Teacher in college" award for the year 2013 & 2014.