



Statistical Evaluation of Task Scheduling Algorithms in Cloud Environments

Vidya Kantale¹, Jignyasa Sanghavi²

¹M.Tech Scholar, Shri Ramdeobaba College of Engineering and Management, Nagpur, India,
kantalevs@rknec.edu

²Assistant Professor, Shri Ramdeobaba College of Engineering and Management, Nagpur, India,
sanghavijb1@rknec.edu

ABSTRACT

Task scheduling algorithms in cloud have come a long way, from simplistic algorithms like first come first serve, to bio-inspired & machine learning algorithms like Q-learning and genetic algorithms. The main objective of any task scheduling algorithm is to minimize the number of execution cycles needed to completely and effectively execute a given set of tasks. In this work, we present a comparison of different scheduling algorithms and their performance evaluation. The proposed research takes into consideration different nuances of cloud task scheduling, which include task length, task completion time, virtual machine configuration and task deadline. Moreover, the review also includes containers, which ensures that the scheduler works effectively on a container environment to improve efficiency of the cloud deployment. We propose different comparisons of the algorithms which are under different task conditions. In our analysis, we reviewed all the algorithms on different tasks and under different virtual machine (VM) configurations. As per the research, the machine learning based algorithms perform better in terms of overall task scheduling efficiency when compared with others.

Key words: Efficiency, Machine, Scheduling, Task.

1. INTRODUCTION

Scheduling of tasks for a cloud computing environment specifically means that the tasks which are being handled by the cloud, must be allocated to proper processing units of the cloud. Various techniques and algorithms are proposed for this in the past, and most of these algorithms are not fully capable for processing the tasks in a real time environment, due to the fact that their efficiency is reduced either due to high computational complexity which results in low response time, or due to less Quality of Service (QoS) for task processing, or due to improper utilization of cloud tasks which is a result of non-optimal & high speed computations. The general approach followed by designers of resource

scheduling algorithms is to first evaluate the number of tasks which are needed to be executed, and create a task pool, once the task pool is created, then create a resource pool, which consists of the various tasks to be used for task scheduling, and finally map the resource pool with the task pool. This process is repeated until the task pool is not empty. Performing the mapping between the task pools and the resource pools usually requires a lot of application specific calculations, for example, if the application requires less response time, then the mapping process must consider the task response time as first priority by keeping other scheduling parameters like execution guarantee or deadline at lower priority, and perform the scheduling accordingly, while if the application has relaxed timing requirements, but needs task execution guarantee, then the algorithm will opt for higher computations, and make sure that all tasks are granted guaranteed execution. In this review, we are combining the different aspects of task scheduling and resource allocation using different algorithms, to make sure that the overall QoS of the cloud computing system is evaluated, the QoS includes but is not limited to, response time, end-to-end delay, task throughput and resource utility factor. The general block-diagram for task scheduling can be explained in figure-1.

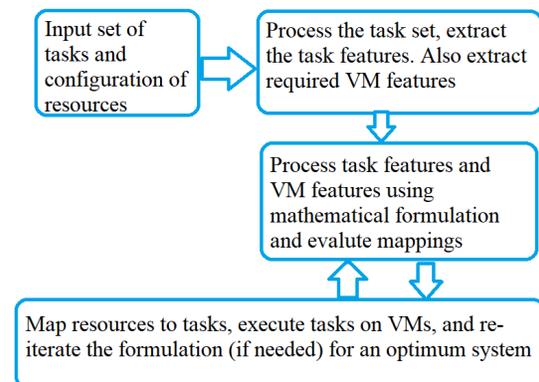


Figure-1: General task allocation procedure

From the figure 1 it is clear that the main block in task scheduling is where processing of task features and mathematical formulation is performed. The next section

compares various algorithms used for task scheduling, and evaluates their performance in terms of common QoS parameters. This comparison allows us to identify some of the best algorithms which are used for task scheduling purposes. Based on the QoS parameters many researchers have proposed trust models that make use of one or more of these parameters. These models are discussed in the next section, with the performance analysis of the models is made in the same section. Finally, we conclude this text with some interesting observations about these models, and suggest some future research that can be done based on these models.

2. LITERATURE REVIEW

In this section different algorithms for resource scheduling are studied, and the selection process of our base algorithm is explained. Resource scheduling requires careful planning from the cloud provider side and from the cloud infrastructure side as well. This planning is accomplished with the help of different task-type-aware algorithms, which take into consideration various task parameters in order to execute the tasks on the resources (or virtual machines). The algorithm defined in [1] uses a modified version of genetic algorithm to schedule resources. They have used simulated annealing with a multi-population genetic algorithm in order to improve task scheduling efficiency. Due to simulated annealing the algorithm focuses on finding global optimum rather than local optimum. Therefore, the overall response time reduces, the completion cost reduces, the convergence speed improves and the degree of load balancing reduces. As per their observations, the completion time reduces by 40%, the load imbalance reduces by 20%, the completion cost reduces by 40% and the load imbalance reduces by 20% when compared with simple genetic algorithm. Due to these advantages, the simulated annealing algorithm can be used for real-time cloud deployments. A different approach towards resource scheduling is proposed in [2], wherein the CPU, memory and bandwidth are used together for all the virtual and physical machines in order to allocate the tasks to the machines. A migration policy is used in order to migrate the tasks which do not satisfy a given allocation constraint, like maximum execution delay, minimum run-time, etc. The proposed algorithm is compared with ZHZJ, ZHCJ and random allocation. The comparison results indicate that the algorithm in [2] reduces the imbalance level of the tasks, and also has a lower running time when compared with ZHZJ and ZHCJ algorithms. Random algorithm does not perform any complex mathematical operations while mapping, thus it is bound to require less delay. In our recommendation, we would suggest that the algorithm in [2] must be compared with more algorithms and larger datasets in order to evaluate its real-time performance. Another bio-inspired algorithm like [1] is mentioned in [3], which uses improved differential evolution algorithm for resource scheduling. The algorithm in [3, 22] proposes the scheduling in a multi-user and multi-provider environment (MUMP). This MUMP environment makes sure that the allocation is done in

real-time, because for any use case scenario there are always multiple providers and multiple users of the cloud deployment. The algorithm is compared with round robin, min-min and differential evolution techniques. The comparisons indicate that the proposed MUMP based algorithm reduces the completion time by 30% and improves the user-to-provider satisfaction ratio by 20%. All this is done while keeping the load ratio of the virtual machines almost constant. For any practical use case, this algorithm can be applied without any modification.

Bio-inspired algorithms can be applied together in order to optimize one-particular area of task scheduling. In the work proposed by researchers in [4], a hybrid algorithm which combines cat swarm optimization for load balancing and particle swarm optimization for virtual machine configuration management is defined. Via their algorithm individual optimizations are combined to form a bigger system-level optimization. Their results when compared with simple PSO, round robin, simple cat swarm optimization, ant-colony optimization and exact optimization showcase a 10% reduction in processing delay, 20% improvement in resource utilization and 10% reduction in algorithm complexity. This work in [4] is a perfect example of layered-based modular processing, which tends to improve the overall efficiency of resource allocation. PSO is a very widely accepted optimization algorithm which has been in use for quite a long time. PSO allows for multiple level modifications, right from its fitness function, to its velocity equations. One such modification is done in [5, 20], where the fitness and velocity update equations are optimized in order to minimize the response time of the algorithm. The modified PSO reduces the response time of the algorithm by more than 10%, and can be used for fast-scheduling of resources in the cloud. Heuristics based scheduling is proposed using the max-min algorithm in [6], wherein researchers have applied maximum sized tasks to minimum capacity virtual machines. By doing this, the minimum sized tasks execute on higher capacity virtual machines, thereby the processing speed of the system improves drastically. The proposed max-min algorithm changes the concept a bit further, by executing average-sized tasks on slower virtual machines, which further reduces the response time by 10% when compared to the original max-min algorithm. A hybrid approach which combines max-min and min-max algorithms is defined in [7], wherein the elapsed time for both max-min and min-max is evaluated, and then based on the comparison a particular algorithm is selected for scheduling. The results indicate that the proposed combinatorial approach reduces the make-span or response time by 10%, when compared with max-min, min-max, RASA, improved-max-min and enhanced version of max-min. These results showcase the superiority of using a pre-calculated metric, rather than fixating on a given policy. Value of service (VoS) is a novel metric defined in [8], wherein the value of the services given by a cloud provider are decided by the virtual machine's previous ability to complete a task within a given deadline and the energy needed by the VM to execute the given task. Using these parameters, a VoS

based scheduling algorithm is defined in [8], which uses soft and hard thresholds for scheduling tasks. The VoS based algorithm is compared with a non-VoS based system, and it is observed that the VoS based system is 50% more responsive, and 40% more energy efficient. Generally, the VoS system is deployed on each of the load-execution units, which increases the overheads. In order to reduce these overheads, a central load balancer is defined in [9]. This balancer uses the concept of data-aggregation in order to execute tasks on virtual machines. The information from all the computing nodes is aggregated on a single node, and calculations are done based on these readings. These calculations provide a fair idea about the current load scenarios of the system, and thereby allows for proper resource allocation. The central load balancer reduces the response time by more than 60% when compared with round robin, active response time and throttled response time-based balancers.

Clustering is a way to group similar kind data together in order to perform a similar set of operation on it. A clustering-based approach, which uses task grouping based on the cloud's capability to execute a given set of tasks on a given set of machines is proposed in [10]. Based on this clustering, the algorithm is able to reduce the processing delay by 15%, and the processing cost by 70% when compared with a non-clustering approach of resource scheduling. Clustering approach is again used to balance network traffic in [11], wherein the algorithm is not a cloud-based algorithm, but can be linearly used for cloud-based systems. Using the proposed algorithm, the mapping between tasks and execution units can be performed in the same manner as the mapping between nodes and base-stations is done. The results indicate that the proposed algorithm reduces delay by 10% when compared to first-come-first-serve (FCFS) based scheduling systems. The evaluation of this algorithm must be done on cloud networks to evaluate its real-time performance before actual usage. Like FCFS; round robin and least connection methods are also equally good when it comes to task scheduling. The work in [12] combines weighted round robin with weighted least connection algorithms in order to improve the performance of each of the individual algorithms. The proposed algorithm reduces the average waiting time by 20%, and increases the average resource utilization by almost 15% when compared with the individual algorithms. This study can further be extended for more algorithms in order to evaluate its actual performance.

Including cost awareness to a scheduling system is of primary importance due to 2 major reasons, which are,

- A cost aware scheduling system will never overload the virtual machines
- The system will always take into consideration Quality of Service while scheduling tasks

Due to these inherent advantages the algorithm in [13] was studied and selected as our baseline algorithm for this text. Using the work in [13], researchers have claimed that cost effective resource scheduling algorithm takes into

consideration the task cost, and allocates the best capacity VMs to the tasks which require higher processing capabilities along with reduced response time. Due to these advantages the resource cost of this CERS algorithm is lower than the existing methods, while the throughput is very high. An in-depth analysis of the CERS system is done in the later sections of this text. In order to evaluate CERS with other algorithms, we have selected the research done in [14, 21]. This research clearly indicates that simulated annealing and bio-inspired algorithms are the best-in-class, and must be used for any level of resource scheduling.

A bandwidth aware scheduling algorithm is presented in [15], wherein researchers have selected the bandwidth of virtual machine in order to allocate tasks. Higher bandwidth machines are assigned with larger tasks, and lower bandwidth machines are assigned with smaller tasks. Due to this bandwidth-based allocation the delay in execution of tasks is reduced by more than 40%. While these results look too-good-to-be-true, we too recommend in-depth analysis of such algorithms before actual implementation on cloud deployments. Another soft-computing algorithm that modifies PSO and adds cuckoo search into it is presented in [16]. Using the cuckoo-based PSO, the cost of task execution reduces by 10%, while the energy efficiency improves by 15%, when compared with simple PSO and improved PSO algorithms. But this algorithm is generally suited for smaller sized tasks, because the complexity of evaluation increases as the task size increases. A Median Deviation based Task Scheduling (MDTS), which uses Median Absolute Deviation (MAD) of the Expected Time to Compute (ETC) is proposed in [14]. In this approach the average of the computation time is evaluated, and then minimized by using task time variation optimization. Due to this approach, the overall response time of the algorithm is reduced by almost 25% when compared to CPOP, HEFT and MDTS algorithms as mentioned in [14]. Clustering based methods are supposed to reduce the overall response time of any scheduling system. This has been proved by the work in [17], where task size and virtual machine capacity-based clustering is performed in order to map the most matching clusters. Clusters with minimum task time are mapped with VM clusters of minimum capacity in order to improve the overall quality of scheduling. This results in a reduction of make span by 10%, an increase in resource utilization by more than 30% and an improvement in trust-level by around 10% when compared with the TTSA algorithm [17]. As seen previously, the bio-inspired algorithms outperform any other statistical approaches. But in contrast the hybrid heuristics method proposed in [18] beats the PSO, ACO and round robin-based approaches in terms of average delay of execution, energy consumption and reliability. Due to the heuristic-based approach, there is no-uncertainty of the obtained solution, thereby there is an obvious improvement in algorithm reliability as compared to other stochastic algorithms, which do not guarantee optimization. Another CERS based algorithm is presented in [19], wherein chemical reaction optimization and ant colony

optimization (CRO and ACO) are used in tandem to improve the effectiveness of task scheduling. Using these two methods, and then adding a resource aware deadline constraint to it, guarantees a reduction in response time, and an improvement in the overall resource utilization by the system.

3. CONCLUSION

Based on the review, we can observe that the non-machine algorithms have longer waiting delays as compared to the machine learning counterparts, the delay is reduced by 8 %. The execution time of the machine learning is improved by 10 % when compared to non-ML technique. Also, the execution time values are found to be lower for ML than for greedy algorithm. Moreover, the mean task waiting time is also reduced by 8 % when compared to the non-ML algorithms. Machine learning in itself is a very computationally advanced process for the problem of load balancing, so there is not much to do on this topic in terms of algorithmic complexity. Researchers can try and implement more advanced techniques like deep nets for solving the issue of load balancing, but the results will be incremental. In order to really optimize the performance further, researchers can use quantum computing for load balancing, and develop quantum computational layers in order to evaluate its performance, and apply the proposed machine learning algorithm on the quantum processors.

REFERENCES

1. Habib Izadkha. **Learning Based Genetic Algorithm for Task Graph Scheduling**, *Hindawi Applied Computational Intelligence and Soft Computing Volume 2019*, Article ID 6543957, February 2019. <https://doi.org/10.1155/2019/6543957>
2. Arghavan Keivani, Farzad Ghayoor, Jules-Raymond Tapamo. **A Review of Recent Methods of Task Scheduling in Cloud Computing**, *2018 19th IEEE Mediterranean Electrotechnical Conference*, June 2018. <https://doi.org/10.1109/MELCON.2018.8379076>
3. Abdul Razaque, Nikhileshwara Reddy Vennapusa, Nisargkumar Soni, Guna Sree Janapati, khilesh Reddy Vangala. **Task Scheduling in Cloud Computing**, *2016 IEEE Long Island Systems, Applications and Technology Conference*, June 2016. <https://doi.org/10.1109/LISAT.2016.7494149>
4. Naresh T, A Jaya Lakshmi, Vuyyuru Krishna Reddy. **Resource Optimization Using Cloud Scheduling**, *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*. ISSN: 2278-3075, Vol-8 Issue-6S2, April 2019.
5. Aida A. Nasr, Nirmeen A. El-Bahnasawy, Gamal Attiya, Ayman El-Sayed. **Cost-Effective Algorithm for Workflow Scheduling in Cloud Computing Under Deadline Constraint**, *Arabian Journal for Science and Engineering*, November 2018.
6. Ebtessam Alobouda, Heba Kurdi. **Cuckoo-inspired Job Scheduling Algorithm for Cloud Computing**, *Procedia Computer Science 2019 – Elsevier*, 2019. <https://doi.org/10.1016/j.procs.2019.04.153>
7. Li Jingmei, Liu Jia, Wang Jiayang. **An Improved Differential Evolution Task Scheduling Algorithm Based on Cloud Computing**, *2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science*, December 2018. <https://doi.org/10.1109/DCABES.2018.00018>
8. Priya Gupta, Makrand Samvatsar, Upendra Singh. **Cloud Computing Through Dynamic Resource Allocation Scheme**, *2017 International Conference on Electronics, Communication and Aerospace Technology*, December 2017. <https://doi.org/10.1109/ICECA.2017.8212723>
9. P Krishnadoss, P Jacob. **OCSA: Task Scheduling Algorithm in Cloud Computing Environment**, *International Journal of Intelligent Engineering and Systems*, February 2018. <https://doi.org/10.22266/ijies2018.0630.29>
10. Ruba Abu Khurma, Heba Al Harasheh, Ahmad Sharieh. **Task Scheduling Algorithm In Cloud Computing Based On Modified Round Robin Algorithm**, *Journal of Theoretical and Applied Information Technology*, September 2018.
11. Babur Hayat Malik, Mehwashma Amir, Bilal Mazhar, Shehzad Ali, Rabiya Jalil, Javaria Khalid. **Comparison of Task Scheduling Algorithms in Cloud Environment** *International Journal of Advanced Computer Science and Applications*, 2018.
12. Weiwei Lin, Gaofeng Peng, Xinran Bian, Siyao Xu, Victor Chang, Yin Li. **Scheduling Algorithms for Heterogeneous Cloud Environment: Main Resource Load Balancing Algorithm and Time Balancing Algorithm**, *Journal of Grid Computing*, October 2019.
13. Tahani Aladwani. **Impact of Selecting Virtual Machine with Least Load on Tasks Scheduling Algorithms in Cloud Computing**, *Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, March 2017. <https://doi.org/10.1145/3090354.3090367>
14. Mahendra Bhatu Gawali, Subhash K. Shinde. **Task scheduling and resource allocation in cloud computing using a heuristic approach**, *Journal of Cloud Computing: Advances, Systems and Applications*, February 2018.
15. Nidhi Tyagi, Vinita Sharma. **Optimized Task Scheduling Algorithms in Distributed Computing: A Review**, *International Journal of Computer Applications*, January 2019. <https://doi.org/10.5120/ijca2019918345>
16. Songtao Guo, Jiadi Liu, Yuanyuan Yang, Bin Xiao, Zhetao Li. **Energy-Efficient Dynamic Computation**

- Offloading and Cooperative Task Scheduling in Mobile Cloud Computing**, *IEEE Transactions on Mobile Computing*, April 2018.
17. Pham Phuoc Hung, Golam Alam, Nguyen Hai, Quan Tho, Eui-Nam Huh. **A Dynamic Scheduling Method for Collaborated Cloud with Thick Clients**, *International Arab Journal of Information Technology*, July 2019.
 18. PeiYun Zhang, MengChu Zhou. **Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy**, *IEEE Transactions On Automation Science And Engineering*, 2017.
 19. HamzaDjigal, JunFeng, JiaminLu. **Task Scheduling for Heterogeneous Computing using a Predict Cost Matrix**, *International Conference on Parallel Processing*, 2019.
<https://doi.org/10.1145/3339186.3339206>
 20. Syed.Karimunnisa, Dr.Vijaya Sri Kompalli. **Cloud Computing: Review on Recent Research Progress and Issues**, *International Journal of Advanced Trends in Computer Science and Engineering*, I ISSN 2278-3091, Vol-8, No.5, March – April 2019.
<https://doi.org/10.30534/ijatcse/2019/18822019>
 21. Dr.S.Sudhakar, Dr.N.Satheesh, Dr.S.Balu, Amireddy Srinish Reddy, Dr.G.Murugan. **Optimizing Joins in a Map-Reduce for Data Storage and Retrieval Performance Analysis of Query Processing in HDFS for Big Data**, *International Journal of Advanced Trends in Computer Science and Engineering*, I ISSN 2278-3091, Vol-8, No.5, September - October 2019.
<https://doi.org/10.30534/ijatcse/2019/33852019>
 22. G. Kiruthiga, Dr. S. Mary Vennila. **An Enriched Chaotic Quantum Whale Optimization Algorithm Based Job scheduling in Cloud Computing Environment**, *International Journal of Advanced Trends in Computer Science and Engineering*, I ISSN 2278-3091, Vol-8, No.4, July – August 2019