

Energy-Aware Cloud Task Scheduling with Medoid PSO Clustering

Posted on October 2, 2026

Abstract

Cloud computing services are increasing day by day in recent times. Especially users from all sectors turn to use cloud computing to fast accessing of required services by investing less amount of money in their applications. As the increase of retrieving services from cloud, the job of task scheduler increases. The cloud task scheduler must see that all the requested cloud tasks take less amount of time. The proposed work is aimed at reducing makespan time of cloud tasks with minimum [energy consumption](#) using the new method Modified Greedy Medoid PSO Clustering (MGMPC). The MGMPC method generated significant results when compared to existing algorithms G&PSO and MGPSO approaches.

Keywords: cloud, computing, pso, greedy, medoid, clustering.

Introduction

Cloud computing is an emerging distributed technology [1] to offer services online with internet. Knowingly or unknowingly, a number of users have been using and engaged with cloud computing technology in their daily activities. Right from the school children to research associates, all are involved in the cloud technology. Users can share their data, knowledge and exchange their views across globe using cloud technology. People need not investment huge amount of money to use this technology. The cloud technology provides pay per use model. Using this, users can utilize services from cloud as per their requirements and pay as per their usage. Especially the pay per use model eases and simplifies tasks of industrialists, organizations, institutions and what not. Mainly the cloud technology reduces burden on It is such a massive and simple technology. The cloud services [2-3] can be software as service, infrastructures as a service and platform as a service. Software as a service enables users to access software applications from cloud by not installing them in their own devices. Examples of software as a service are google apps, Facebook apps, amazon apps, WhatsApp, etc. All the

software apps and applications are accessed by enabling internet services in their devices. Infrastructure as a service enables users to access infrastructure from a cloud. Providing infrastructure is in the form of proving servers, secondary storages, RAM storages, networking etc. Users can simply opt packages of infrastructure according to their needs. Platform as a service enables users to access platform from a cloud. Providing platform is in the form of arranging facilities

for deploying user's applications, providing operating systems etc. When users make requests to access services from cloud, they are treated as cloud tasks. To get faster response to user's requests, cloud tasks must be executed very fast. So the cloud task scheduler monitors execution of cloud tasks by having knowledge of its cloud resources like virtual machines. The order of cloud tasks execution in virtual machines impacts the overall response time. Hence it is always to see that makespan of cloud tasks execution must be minimum. There are a lot of algorithms available for [cloud task scheduling](#) [4]. They are preemptive & non-preemptive scheduling algorithms, static & dynamic algorithms, heuristic algorithms etc. Heuristics algorithms are very much useful to provide approximation results to cloud applications of kind NP-hard problems. The proposed work concentrates on greedy based PSO heuristic approach with clustering concept to minimize cloud tasks makespan time. K-medoid technique has been used for clustering concept in our work. The organization of the remaining paper is as follows. Section 2 shows the literature survey related to task scheduling work. Working nature of proposed methodology is presented in section 3. Experimental results are discussed in section 4. At last section 5 presents the conclusion.

Literature Survey

Task scheduling plays a major role in minimizing the makespan of cloud tasks execution time. Heuristic algorithms give better results in the form of approximating the objective functions like minimizing makespan. A cloud is a type of grid like architecture where distribution of resources is allowed. A grid has a number of resources which are to be distributed across the systems. A better scheduling helps to reduce the total distribution time of resources. [5] Z. Pooranian, et al. introduced a gravitational emulation local search algorithm with the combination of PSO algorithm to reduce makespan in grid computing. Heuristic algorithms like genetic algorithms can also be used for task scheduling. But PSO algorithm can produce better results when compared to genetic algorithms for task scheduling especially in grid computing. [6] L. Zhang, et al. proved the previously said statement. PSO algorithm has a nature of bird flocking behavior. This nature helps to have good collaborative approach for search of optimal solution. In distributed computing, the PSO algorithm produces good results compared to genetic algorithms. [7] A. Salman, et al. proved the previously said sentence. [8] Solmaz Abdi, et al. used the PSO algorithm with some modifications to reduce makespan in the cloud computing environment. [9] Awad, et al. showed how a PSO algorithm can be enhanced to generate better makespan results in cloud environment. The authors incorporated load balancing approach for better load balancing resources in the cloud. [10] Omara, et al. showed a simple PSO algorithm for task scheduling in cloud environments. [11] Vamsi Krishna, et al. used a dynamic PSO algorithm for task scheduling. Cuckoo search algorithm was used with the PSO technique in

their work. The algorithm works better for heterogeneous workloads. Load balancing and heuristic algorithms can be combined to generate efficient results in cloud environments for task scheduling. [12] Rafiqul Zaman Khan, et al. showed how an adaptive load balancing technique was used with PSO for generating effective makespan results. [13] Gaurav Sharma, et al. showed the

minimization of makespan by categorizing the tasks of similar into one group. The authors modified the PSO algorithm for generating good results. Clustering techniques can be incorporated in task scheduling algorithms for cloud environments to minimize makespan. Above all the approaches have not emphasized on using clustering techniques for task scheduling concept. The most widely used clustering techniques are partitioning techniques. K-medoid techniques [14-15] generate good results compared to k- means techniques as they take medoid as parameter for clustering. The authors Nagaraju Devarakonda, et al. [16-18] showed how clustering techniques can be implemented to generate better makespan results in cloud environment. The proposed paper is aimed at minimizing makespan and energy consumption in the cloud environment with clustering techniques by introducing the new method MGMPC. The MGMPC algorithm makes use of PSO algorithm by modifying the existing G&PSO algorithm. The MGMPC algorithm uses k-medoid algorithm for clustering.

Proposed Method

The block diagram of proposed method is MGMPC shown in the below figure. The MGMPC is abbreviated as modified greedy medoid PSO clustering which uses the benefits of k-medoid and PSO techniques. Initially the cloud tasks are taken as input to the algorithm. The taken cloud tasks are given to the k-medoid clustering algorithm. The k-medoid algorithm is used to cluster the given cloud tasks into two groups namely high performance and low performance groups. K-medoid algorithm clusters the cloud tasks in a better way compared to k-means algorithm. Low performance clustered groups are assigned to low performance virtual machines and high performance cluster groups are assigned to high performance virtual machines in a cloud environment. Now each cloud task in the clustered group is sorted in descending order according to their performances. Similarly, virtual machines are also sorted in descending order. Greedy approach is used for initial allotment of cloud tasks to virtual machines as follows. For each clustered group, the assignment of cloud tasks to virtual machines is done as the first cloud task is assigned to first virtual machine and the next cloud task is assigned to next highest performance virtual machine in the cloud environment. This process goes on until all the cloud tasks are assigned to virtual machines. The resultant scheduling list and makespan is taken as initial global solution for PSO algorithm. Now the PSO algorithm is applied to all clustered groups which use greedy approach. After applying PSO algorithm, the final scheduling list and makespan is taken as final solution out approach. The proposed approach took the less energy consumption when compared to the existing techniques.

Figure 1. Block diagram of MGMPC.

Results and Discussion

The experimental results were obtained using cloudsim tool. A total number of hundred cloud tasks were considered for results. Cloud task lengths varied from 500 MI to 1000 MI. Five virtual machines were used for experimental results where their performances vary from 500MIPS to 900 MIPS. There

virtual machines were grouped into two categories named as low performance and high performance virtual machines based on their performances. The population size considered for PSO algorithm was 100. Now the cloud tasks were given to proposed algorithm and tested makespan time with the above parameters in the cloud environment. The resultant makespan time values were compared with the existing methodologies. The below table 1, and fig. 2 show the comparison of makespan time.

Table 1. Makespan time obtained for 100 tasks

Methodology	Makespan time
PSO [18]	17.92
G&PSO [18]	17.8
MGPSO [18]	13.83
MGMPC [proposed]	12.61

MGMPC workflow: K-Medoid partitioning separates tasks into high-performance and low-performance cluster groups. Tasks are sorted, processed through the greedy method and PSO technique, and then mapped to virtual machines.

The above figure clearly states that proposed method MGMPC obtains very less makespan time compared to existing methodologies PSO, G&PSO and MGPSO. In terms of performance in percentages, the proposed method showed improvement of 29.63%, 29.15%, and 8.82% over PSO, G&PSO and MGPSO respectively. The loads on all virtual machines was also tested and showed in the below table 2 and fig. 3.

Table 2. Load distributed on five virtual machines in percentages

Methodology	VM1	VM2	VM3	VM4	VM5
PSO	33.33	33.33	33.33	43.47	56.52
G&PSO	28.57	35.71	35.71	50.00	50.00
MGPSO	33.33	33.33	33.33	50.00	50.00

Methodology	VM1	VM2	VM3	VM4	VM5
MGMPC	33.33	33.33	33.33	47.82	52.17

Load parameter is considered as the number of tasks distributed to virtual machines. Above figure represents load distribution in percentages. Here the virtual machines VM1, VM2, VM3 were considered as low performance machines and VM4, VM5 as high performance machines. The performance constraint used for virtual machines were $VM1 < VM2 < VM3$ and $VM4 < VM5$. To obtain better results low performance machines should be assigned less number of tasks compared to high performance machines so that distribution of load will not be more on low performance machines. The above table 2 and fig. 3, clearly states that there was a more balanced load distribution among 5 virtual machines in proposed approach compared to existing methodologies.

The proposed methodology was also tested using power consumption in the cloud environment. Power consumption for a host is calculated as:

$$p = k \times \text{max} + (1 - k) \times \text{max} \times \text{utilization} \quad (1)$$

where k is the power utilization in the idle state, max is the maximum power utilization of a host, and utilization is the average power utilization by a host. Power consumption was tested on the standard CloudSim workload data [19], using sample workload data from 1,160 computers across 25 nations and 547 stations.

Table 3. Comparison of power consumption in kW/hr

Methodology	Min	Max
IQR_MMT [19]	90	139
IQR_RS [19]	96	139
IQR_MU [19]	94	137
PS-ABC [19]	92	129
PS-ES [19]	90	121
DataABC [19]	80	120

Methodology	Min	Max
IGSA [19]	59.8	72.8
MGMPC [proposed]	51.5	65.3

From Table 3, the proposed method shows lower minimum and maximum power-consumption values than the compared methodologies.

Conclusion

Cloud task scheduling is very important one which is used to schedule tasks in the cloud environment. Better the schedule tasks, lesser the makespan time which is required to execute the cloud tasks in a less amount of time. Reduction of power consumption and load balancing is also very much needed while scheduling. Hence the inclusion of clustering techniques in the bio- inspired heuristic techniques derives better results while addressing the said issues. The proposed approach MGMPC is the one which showed good results in terms of giving lesser makespan, minimizing the energy consumption and better load balancing with the combination of clustering techniques & bio-inspired heuristic techniques.

References

1. Hoefer, C. N., and G. Karagiannis. 2010. "Taxonomy of Cloud Computing Services." 2010 IEEE Globecom Workshops, GC '10, 1345-50.
2. Akilandeswari, P., and H. Srimathi. 2016. "Survey and Analysis on Task Scheduling in Cloud Environment." Indian Journal of Science and Technology 9 (37). <https://doi.org/10.17485/ijst/2016/v9i37/102058>.
3. P. NagendraBabu, M. ChaitanyaKumari, and S. Venkata Mohan. 2018. "A Literature Survey on Cloud Computing." I-Manager's Journal on Information Technology 1 (1): 44-49. <https://doi.org/10.26634/jit.1.1.1842>.
4. Neha Sharma and Sanjay Tyagi. 2016. "Task Scheduling in Cloud Computing." Advances in Computer Science and Engineering, 249-252.
5. Z. Pooranian, M. Shojafar, J. H. Abawajy, and A. Abraham. "An Efficient Meta-Heuristic Algorithm for Grid Computing." Journal of Combinatorial Optimization, 2013, pp. 1-22.
6. L. Zhang, Y. Chen, R. Sun, S. Jing, and B. Yang. "A Task Scheduling Algorithm Based on PSO for Grid Computing." International Journal of Computational Intelligence Research 4 (2008): 37-43.
7. A. Salman, I. Ahmad, and S. Al-Madani. "Particle Swarm Optimization for Task Assignment Problem." Microprocessors and Microsystems 26 (2002): 363-371.

8. Solmaz Abdi, Seyyed Ahmad Motamedi, and Saeed Sharifian. 2014. "Task Scheduling Using Modified PSO Algorithm in Cloud Computing Environment." International Conference on Machine Learning, Electrical and Mechanical Engineering, Dubai, UAE, January 8-9, 2014.
9. Awad, A. I., El-Hefnawy, N. A., and Abdel-Kader, H. M. 2015. "Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments." *Procedia Computer Science* 65: 920-929. doi:10.1016/j.procs.2015.09.064.
10. Al-Maamari, A., and Omara, F. A. 2015. "Task Scheduling Using PSO Algorithm in Cloud Computing Environments." *International Journal of Grid and Distributed Computing* 8 (5): 245-256. doi:10.14257/ijgdc.2015.8.5.24.
11. Sudheer, M. S., and Vamsi Krishna, M. 2019. "Dynamic PSO for Task Scheduling Optimization in Cloud Computing." *International Journal of Recent Technology and Engineering* 8 (2 Special Issue 11): 332-338. doi:10.35940/ijrte.B1052.0982S1119.
12. Md Oqail Ahmad and Rafiqul Zaman Khan. 2019. "PSO-Based Task Scheduling Algorithm Using Adaptive Load Balancing Approach for Cloud Computing Environment." *International Journal of Scientific & Technology Research* 8 (11).
13. Neha Miglani and Gaurav Sharma. 2019. "Modified Particle Swarm Optimization Based upon Task Categorization in Cloud Environment." *International Journal of Engineering and Advanced Technology* 8 (4C).
14. Park, H. S., and Jun, C. H. 2009. "A Simple and Fast Algorithm for K-Medoids Clustering." *Expert Systems with Applications* 36 (2 Part 2): 3336-3341. doi:10.1016/j.eswa.2008.01.039.
15. Schubert, E., and Rousseeuw, P. J. 2018. "Faster K-Medoids Clustering: Improving the PAM, CLARA, and CLARANS Algorithms." Available at: <http://arxiv.org/abs/1810.05691>.
16. Y. Home Prasanna Raju and N. Devarakonda. 2019. "Cluster Based Hybrid Approach to Task Scheduling in Cloud Environment." *International Journal of Advanced Computer Science and Applications* 10 (4): 425-429. doi:10.14569/ijacsa.2019.0100452.
17. Y. Home Prasanna Raju and N. Devarakonda. 2020. "A Cluster Medoid Approach for Cloud Task Scheduling." *International Journal of Knowledge-Based and Intelligent Engineering Systems*.
18. Y. Home Prasanna Raju and N. Devarakonda. 2020. "Greedy Based PSO with Clustering Technique for Cloud Task Scheduling." 3rd International Conference on Computational Intelligence & Data Engineering.
19. J. Jiang, Y. Feng, J. Zhao, and K. Li. 2017. "DataABC: A Fast ABC Based Energy-Efficient Live VM Consolidation Policy with Data-Intensive Energy Evaluation Model." *Future Generation Computer Systems* 74: 132-141.