World Wide Journal of Multidisciplinary Research and Development

WWJMRD 2017; 3(11): 327-331
www.wwjmrd.com
International Journal
Peer Reviewed Journal
Refereed Journal
Indexed Journal
UGC Approved Journal
Impact Factor MJIF: 4.25
e-ISSN: 2454-6615

Banita
M. Tech (Student)/PG Scholar Department of Computer Science, Engineering GZS Campus College of Engineering, Technology (MRSPTU), Bathinda, Punjab, India

Shaveta Rani
Professor, Department of Computer Science, Engineering GZS Campus College of Engineering, Technology (MRSPTU), Bathinda, Punjab, India

Paramjeet Singh
Professor, Department of Computer Science, Engineering GZS Campus College of Engineering, Technology (MRSPTU), Bathinda Punjab India

Correspondence:
Banita
M. Tech (Student)/PG Scholar Department of Computer Science, Engineering GZS Campus College of Engineering, Technology (MRSPTU), Bathinda, Punjab, India

Optimization of Scheduling Algorithm in Cloud Computing Using Particle Swarm Optimization

Banita, Shaveta Rani, Paramjeet Singh

Abstract
Cloud is the central collection of types of resources. It exists in two ways it can be private cloud, public cloud. Users who are the member of the cloud can use the cloud services on pay per use basis. Various clients put the request to the cloud for services it provides. The entire request will be submitted to the central virtual machine. Virtual machine to avoid the deadlock situation put the scheduling algorithm on to the requests submitted by the users. So that unnecessary competitions can be avoided. Min-Min and Max-Min algorithms can be failed if the number of resources with maximum requirements is more or number of resources with minimum requirements is more. To avoid this conflict Improved Max Min (existing algorithm) is used which is the modification of the Max-Min algorithm by using the rasa technique. RASA is basically the awareness of the resource i.e. when the resource is free, and in how much time it takes to complete the task etc. In current research PSO as optimization technique is applied to identify the optimum resource for the process. PSO algorithm is implemented onto the existing algorithm i.e. Improved Max-Min algorithm. The proposed algorithm is Immpso (Improved Max-Min Particle Swarm Optimiazation). Then the performance of the proposed algorithm has been compared the existing algorithm on the basis of two different parameters like Power Consumption and Throughput. Scheduling based on a proposed algorithm has better performance as compared to the existing algorithm. In proposed algorithm Power Consumption has improved 17.10% and throughput has improved 10.91% as compared to the existing algorithm.

Keywords: Cloud Computing, Max-Min Algorithm, Min-Min Algorithm, Improved Max-Min algorithm and PSO.

Introduction
Cloud processing is an updated step towards new processing era. Cloud provides the online resources and online storage to the users. Cloud provides whole data at the lower cost. With the help of internet, users can access all the resources in cloud computing at any time. The users who are the part of the system of cloud can use the resources in cost effective way, because it is based upon the pay per use based computing.
The success of the clouds is being measured due to the ability and availability of virtualization technology. By having virtualization single processor can run more than one operating system simulteously. Virtualization provides the ability to efficiently utilizing the hardware and software's.
Cloud can be of two types either private cloud or public cloud. In each type basic objective is to provide the resources to the users at cost effective way.

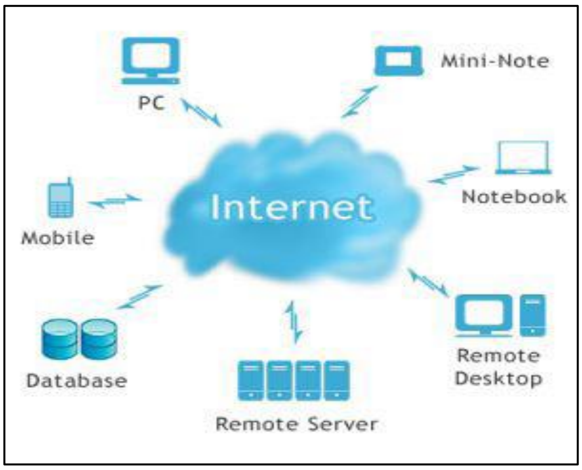


Fig.1: Cloud Computing [14]

Scheduling

The resources collected in the cloud are being put in use for the requests submitted by the user. The request submitted is to be scheduled for the users. Main advantage of the cloud scheduling is to achieve best throughput and better resource utilization. Cloud scheduling can be categorized into three parts one is resource discovery and filtering process, second is resource selection and third is task submission. In resource discovery datacenter broker discover the resource available in cloud. Resource identification is always based on the resource and task criteria. According to the criteria if resource availability is there, then resource selection is taken place. And selected resource will be allocated to the task.

Min-Min Algorithm

Min-Min scheduling is based choosing that tasks which has minimum completion time. In this type of scheduling that resource is selected, that has minimum completion time. The completion time is selected based on the Meta tasks list. The Main goal of Min-Min algorithm is to complete the task in minimum time and minimum waiting time.

1. Collect all the tasks in central pool of tasks Tk.
2. Let all the resources as Rj.
3. Calculate or $Compute_{ij} = Execution_{ij} + Resource_j$
4. While central task list is empty
5. Choose that task which has minimum completion time
6. Allocate that task to the resource having minimum completion time.
7. Remove the task from tasks list.
8. Update the list of resources.
9. Update completion time for each resource.

Max-Min algorithm

It is another way works opposite to the Min-Min. In Max-Min algorithm that resource is located which has maximum completion time. So that, maximum task is allocated to that resource, that has maximum completion time. So that, multiple smaller tasks can be allocated to the smaller completion time resources. It will make easy to execute multiple tasks to be completed concurrently.

1. Build a single central list of tasks.
2. Build a single central list of resources.
3. Compute the completion time for each resource.
4. While central list become empty
5. Find the task from the central pool which has maximum completion time.
6. Assign that task to resource which has maximum completion time.
7. Completed task is removed from the task list.
8. Update the list of resources.
9. Update the completion time for each task.

Improved Max-Min Algorithm

In both Min-Min and Max-Min the scheduling techniques will be failed when there are multiple smaller tasks and multiple larger tasks. In case of multiple larger tasks Max-Min will be failed because in Max-Min algorithm multiple large tasks cannot be allocated to the multiple large completion time. Similarly when there are large numbers of smaller tasks and it is not possible to allocate all the tasks to be allocated to multiple smaller resources. So in RASA scheduling Algorithm, both the algorithms (Max-Min and Min-Min) is performing simultaneously. So that Resource awareness is performed. When there is large no. of larger tasks then max. Completion time will be selected and when there is large no. of smaller tasks then Min-Min is used.

1. Build a central pool of tasks.
2. Keep all the resources as resource list.
3. Compute the completion time for each resource.
4. Do till all the tasks are mapped to the resources.
5. Check if the resources count is even
6. Find the tasks and there expected completion time also finds the resource that maps it.
7. Locate the task which has maximum completion time.
8. Allocate this task to the resource which is fastest in the available list of tasks, so that task can be completed in minimum time.
9. Remove the task from the task list.
10. Update the resource list.
11. Update the task list.
12. Else
13. For each task in Meta task find the expected completion time and the resource that make it.
14. Locate the task which has minimum completion time.
15. Allocate the task to the resource which has maximum Completion time.
16. Remove the task from the task list
17. Update the resource list
18. Update the completion time.
19. End does.

Literature Survey

1. Shuibing He *et al.* [2]: In this paper, they have considered to improve scientific workflows in cloud environments where data transfers between tasks are performed via provisioned in-memory caching as a service, instead of relying entirely on slower disk-based file systems.

2. Devipriya [17]: In this paper the max. Utilization of the resources with max Availability and minimum completion time is selected. This paper has proposed the technique to overcome the problem of Min-Min and Max-Max using resource allocation of selection basis.

3. Rajwinder Kaur *et al*. [16]: Cloud computing requires efficient utilization of resources and also enhance the performance of the system. So that minimum cost can be incurred in on to the resources utilization.

Proposed "Immpso" Algorithm in Cloud Computing

PSO is a dynamic load balancing technique. It can be used in decentralized load balancing. It is used for dynamic decentralized optimum load balancing at broker level.

This proposed algorithm of resource identification based on the availability of optimum resource. So that optimum efficiency can be achieved. This technique is based on Bird flock who visits in all the direction to identify the optimum food grain. So that all the Birds are fly towards the optimum resource. Similarly in this algorithm all the resources are visited for their effectiveness in current situation. That resource is selected which is optimum for the situation. So that maximum Efficiency can be achieved. This proposed algorithm is based genetic behavior of the Bird flock. That they have learned from natural process. This technique is highly efficient.

IMMPSO Algorithm

Step1: In first step set all the initial variables. Like maximum weight, Minimum weight, number of iteration, size of the resource set, number of tasks in the set to be allocated to with the resources.

Step2: Starts the population. Checks the availability and execution time for each resource lies into the set.

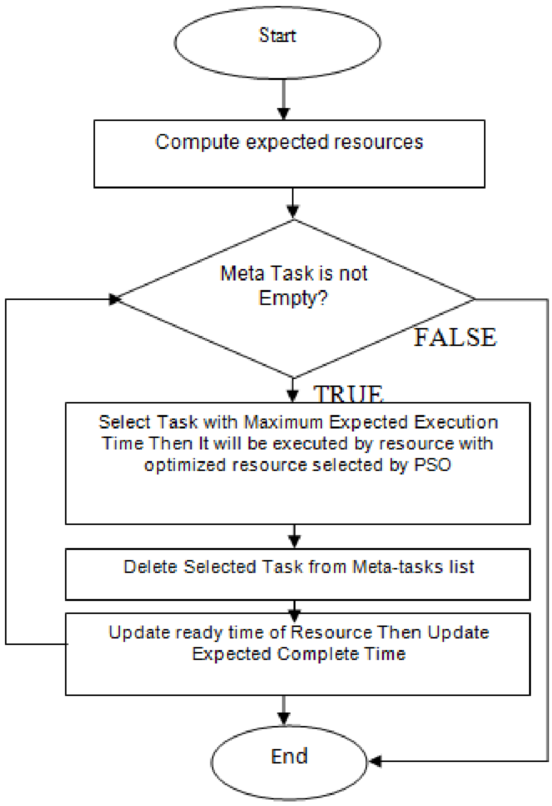
Step3: In this step check the resource required for the current task to be executed. What is the latest execution time for each resource? So that best resource can be identified and allocate the Pbest value.

Step4: Compare the Pbest value of each resource. So that after comparison that resource is selected that is having best Pbest value?

Step5: Obtain the current load of each resource and once load is completed the resources will be removed from the resource list.

Step6: Once the number of iteration is completed for the given situation the procedure will automatically be stopped.

IMMPSO Flow Chart



Analysis

Power Consumption for Data Center 1

This graph shows the comparison for power consumption between improved max-min and proposed algorithm. Clearly it is depicted proposed algorithm is more efficient compared to the improved max-min. So that less energy is required to schedule all the resources to all the processes for the efficiency of the process. It is the total power consumed while executing the task from a gibe resource. In case of proposed algorithm the power consumption is less as compare to the improved max-min. That means in context to the power, the performance of proposed algorithm has improved to 17.10%.

Average power of Improved MAX-MIN	Average power of Proposed Algorithm
102.3881	84.87

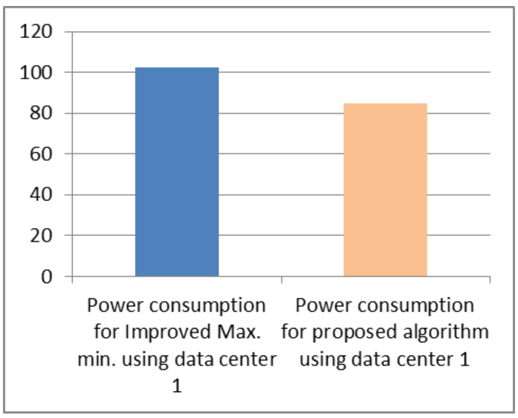


Fig.1: Average Power Consumption of Improved Max-Min and Proposed algorithm

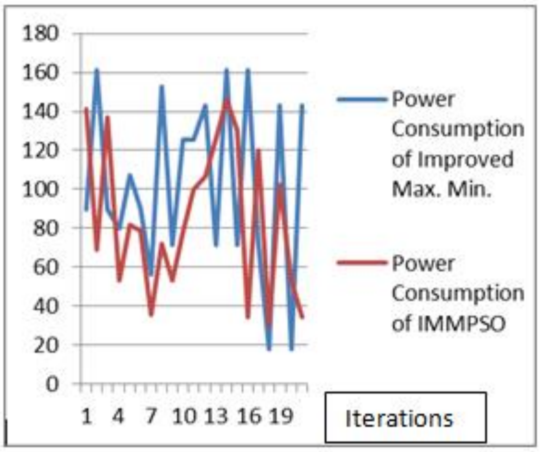


Fig.2: Power Consumption of Improved Max-Min and Proposed algorithm

Power Consumption for Data Center 2

This graph shows the comparison for power consumption between improved max-min and proposed algorithm. Clearly it is depicted proposed algorithm is more efficient compared to the improved max-min. So that less energy is required to schedule all the resources to all the processes for the efficiency of the process. It is the total power consumed while executing the task from a gibe resource. In case of proposed algorithm the power consumption is less compare to the improved max-min. That means in context to the power, the performance of proposed algorithm has improved to 8.48%.

Average power of Improved MAX-MIN	Average power of Proposed Algorithm
92.19	84.37

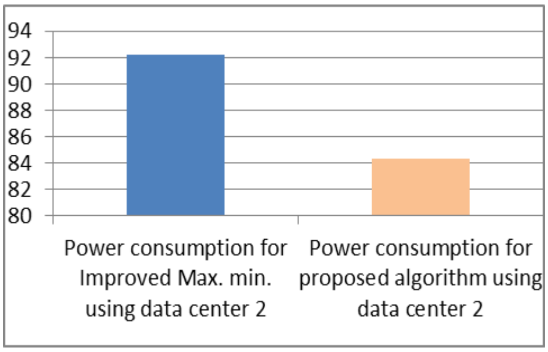


Fig.3: Average Power Consumption of Improved Max-Min and Proposed algorithm

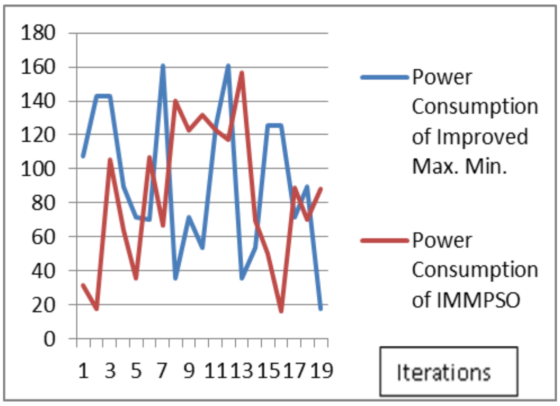


Fig.4: Graph between Improved Max-Min and proposed algorithm.

Throughput for Data Center 1

This graph depicts the throughput comparison between improved max-min and proposed algorithm. Throughput has improved over to the improved max-min. it is the performance parameter in terms to number of processes per unit interval of time. In case the proposed algorithm the throughput has improved to 10.01%.

Average throughput of Improved MAX-MIN	Average throughput of Proposed Algorithm
17.12	18.85

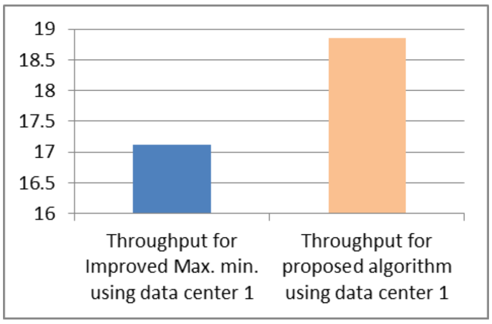


Fig.5: Average Throughput of Improved Max-Min and proposed algorithm

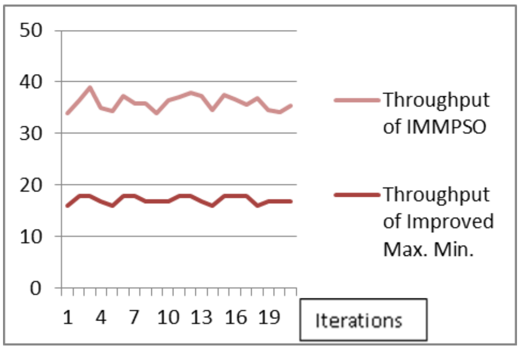


Fig.6: Graph of Improved Max-Min and proposed algorithm

Throughput for Data Center 2

This graph depicts the throughput comparison between improved max-min and proposed algorithm. Throughput has improved over to the improved max-min. It is the performance parameter in terms to number of processes per unit interval of time. In case of proposed algorithm the throughput has improved to 11.21%.

Average throughput of Improved MAX-MIN	Average throughput of Proposed Algorithm
16.62	18.48

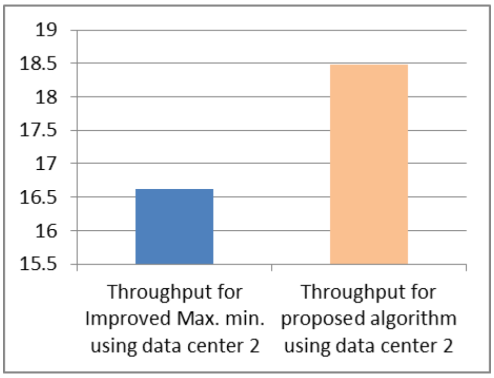


Fig.7: Average Throughput of Improved Max-Min and proposed algorithm

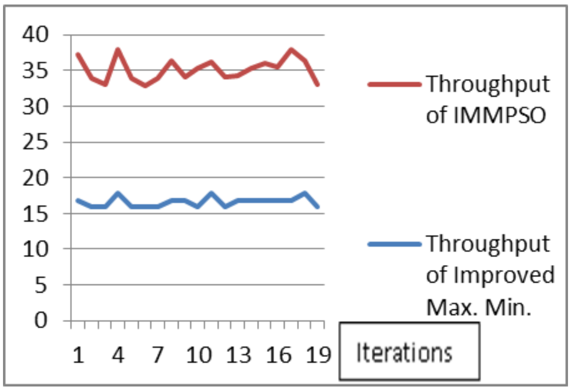


Fig.8: Graph of Improved Max-Min and proposed algorithm

Conclusion

From the current research it is clear that cloud efficiency will depends upon this issue that how well cloud schedules the resources amongst different processes. MAX-MIN and MIN-MIN individually are not so efficient because there may be various longer or even shorter tasks. For optimization of the selection process in current research we have used PSO on improved Max-Min. This technique identifies the best possible resources amongst the multiple available resources. In previous research the RASA based technique was used i.e. improved max-min. Performance parameters like throughput; power consumption has been used to compare the performance of previous and current research. Proposed algorithm has improved upon the power consumption and throughput. Power Consumption has improved 17.10%. And Throughput has improved upon 10.91%.

Future Work

In current research IMMPSO is implemented for maximize the resource utilization and also to minimize the time of execution. In future other genetic based techniques can be applied so that global best technique for further execution time improvement can be identified.

References

1. Neeta Patil, "A review –different scheduling algorithms in cloud computing environment", 11th International Conference on Intelligent Systems and Control (ISCO), 2017.
2. Shuibing He, Yang Wang, Xian-He Sun IEEE Fellow and Chengzhong Xu, "Using Min-Max-Memory Claims to Improve In-Memory Workflow Computations in the Cloud", 1045-9219 (c) 2016 IEEE.
3. Huili Chen et al, "A New Heuristic scheduling strategy LBMM in cloud computing," 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2016 IEEE
4. Vatsal Gajera, "An Effective Multi-Objective Task Scheduling Algorithm using Min-Max Normalization in Cloud Computing," 2016 IEEE.
5. KanwarPreet Kaur, "A Hybrid Approach of Load Balancing through VMs using ACO, Min-Max and Genetic Algorithm," 2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016) Dehradun, India 14-16 October 2016
6. R. Durga Lakshmi et. al. "A Dynamic Approach To Task Scheduling In Cloud Computing Using Genetic Algorithm," 20th March 2016. Vol.85. No.2
7. Long Thai, Blesson Varghese and Adam Barker, "Task Scheduling on the Cloud with Hard Constraints", 2015 IEEE World Congress on Services.
8. Dr.M.Sridhar, "Hybrid Particle Swarm Optimization Scheduling for Cloud Computing", 2015 IEEE.
9. Santhosh B, "A Hybrid Avg. Task-Min and Max-Min Algorithm for Scheduling Tasks In Cloud Computing," 2015 IEEE.
10. Michael Pantazoglou, "Decentralized and Energy-Efficient Workload Management in Enterprise Clouds," 2015 IEEE
11. Navtej Singh et al, "Dynamic Combination of Improved Max-Min and Ant Colony Algorithm for Load Balancing in Cloud System," 6th ICCCNT - 2015 July 13 - 15, 2015,
12. Eakta Kumari and Monika, "A review on task scheduling algorithms in cloud computing", International Journal of Science, Vol. 4, No 2, 2015, 433 – 439.
13. R.Durga Lakshmi1, N Srinivasu, "A Review and Analysis of Task Scheduling Algorithms in Different Cloud Computing Environments" International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 4, Issue. 12, December 2015, pg. 235 – 241.
14. Garima Mahajan, "Job Scheduling In Cloud Computing: A Review of Selected Techniques" International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 3, Issue 3, May-June 2014.
15. Alban Allkoci1, "Comparing Priority and Round Robin Scheduling Algorithms," IJCSI International Journal of Computer Science Issues, Vol. 11, Issue 3, No 1, May 2014
16. Rajwinder Kaur and Pawan Luthra, "Load Balancing in Cloud Computing", Association of Computer Electronics and Electrical Engineers, 2014.
17. S. Devipriya, "Improved max-min heuristic model for task scheduling in cloud", 978-1-4673-6126-2/13.
18. Shaminder Kaur and Amandeep Verma, "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment" I.J. Information Technology and Computer Science, 2012, 10, 74-79.