ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)

on 21st & 22nd April 2014, Organized by

Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

An Improved Task Scheduling Algorithm based on Max-min for Cloud Computing

Santhosh B[1], Dr. Manjaiah D.H [2]

Research Scholar, Dept. of Computer Science, Mangalore University, Karnataka, India[1]

Professor, Dept. of Computer Science, Mangalore University, Karnataka, India[2]

ABSTRACT: In this paper a unique modification to the Improved Max-min algorithm is proposed. In Improved Max-min algorithm largest job is selected and assigned to the resource which gives minimum completion time. Here two algorithms are proposed on Improved Max-min where instead of selecting the largest task , a task just greater than average execution time is selected and assigned to the resource which gives minimum completion time. The experimental results shows the new algorithms schedules jobs with lower makespan .

KEYWORDS: cloud computing, Max-min algorithm, Improved Max-min Algorithm, makespan, minimum completion time, minimum execution time

1. INTRODUCTION

Cloud computing has become a new age technology that has got huge potentials in enterprises and markets. Cloud computing now is known as a provider of dynamic services using very large scalable , on demand, virtualized resources over the internet. It also make it possible to access applications and associated data from anywhere. Companies are able to rent resources from cloud for storage and other computational purposes so that their infrastructure cost can be reduced significantly. They can also make use of software's, platforms and infrastructures as a services , based on pay-as-you-go model.

Some of the drawbacks of the cloud computing is data centre network structure expansibility, energy conservation, replica policies, security and scheduling mechanism [1] [2] [3]. The primary objective of this paper is optimizing the scheduling polices. In cloud computing optimum scheduling is related to optimizing the resources being allocated. Because of the uniqueness of the model, resource allocation is performed with the objective of minimizing the costs associated with it. To make a set of cloud services an effective efficient provider infrastructure one of its requirements is an effective Task scheduling algorithm. Task scheduling algorithm is responsible for mapping jobs submitted to cloud environment onto available resources in such a way that the total response time ,the makespan is minimized [4]

One of the feature of the Max-min algorithm is it selects the largest job and is executed on the fastest available resource. The main drawback of the algorithm is it delays the execution of the smaller jobs and because of the dynamic nature of the cloud , execution of the smaller jobs may be postponed indefinitely. Solution to this is improved Max-min , it works well for the given set of jobs but dynamic cloud environment where the jobs are submitted at any time the algorithm results in degradation in the performance.

In this paper we are proposed two algorithms to increase the efficiency of improved Max-min. The remaining part of the paper are organized as follows : section II presents some related works on Max-min algorithm. Section III presents our proposed modified two algorithms on Improved Max-min algorithm , Section IV describes the Experimental results where we compare the result of the proposed algorithm with other Max-min algorithms. Finally Section V concludes the paper and presents future work.

II. RELATED WORKS

2.1 Max-min Algorithm:

The Max-min Algorithm is based on "select a task with maximum execution time and assign to the resource with minimum execution time"

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)

on 21st & 22nd April 2014, Organized by

Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

2.2 Improved Max-min Algorithm[5]

The Max-min Algorithm is based on "select a task with maximum execution time and assign to the resource which gives minimum completion time"

2.3 Enhanced Max-min Algorithm [6]

Some times in a mata-task largest task is too large compare to the other tasks . It may result in increase in the overall makespan . Here a solution is proposed, first by selecting the job just greater than the average execution time and assigning to the slowest resource and then executing the improved Max-min algorithm.

Max-min Algorithm gives priority to the larger tasks , first it assigns the time consuming jobs to the resources. The main disadvantage of this algorithm is early execution of the large task might increase the total response time of the system. A solution to this is the improved Max-min algorithm which is based on the completion time. Further the result is improved in Enhanced Max-min algorithm first executing largest job in the slowest resource and then using the improved Max-min. In the dynamic environment of the cloud, jobs are submitted batch wise. Always first selected largest job is the largest job within the batch , when there are multiple batches of job this algorithm does not give the efficient result.

III. PROPOSED ALGORITHM

The proposed algorithm is the improvement over to the Improved Max-Min algorithm [5], here the initial step of Enhanced Max-Min Task scheduling algorithm [6] is considered. In Enhanced Max-Min ,first largest task ,just greater than the average execution time is selected and is assigned to the slower resource then the Improved Max-min algorithm is followed. Our proposed algorithm is the modification of the Improved Max-min algorithm where the Largest task just greater than the average execution time is selected every time . Here we are proposing two algorithms In algorithm 1 average execution time is calculated using the arithmetic mean and in algorithm 2 geometric mean is used. Here arithmetic mean and geometric means are used because if the values are independent then arithmetic mean is the best average. If the values are dependent on other values then the geometric mean is the best average. In cloud environment jobs may be independent or dependent of each other and completion time of the job depend on the completion time of the other jobs.

3.1 Improved Algorithm 1 on Max-Min Task Scheduling Algorithm (proposed)

Algorithm:

1. For all submitted tasks in Meta-task; Ti
1.1. For all resources; Rj
1.1.1. Cij = Eij + rj
2.While Meta-task not Empty
2.1 Find task Tk costs Arithmetic Average or nearest Greater than Arithmetic Average execution time.
2.2 Assign task Tk to resource Rj which gives minimum completion time (Slowest resource).
2.3. Remove task Tk from Meta-tasks set.
2.4. Update rj for selected Rj.
2.6. Update Cij for all j.

3.2 Improved Algorithm 2 on Max-Min Task Scheduling Algorithm (proposed)

Algorithm:

1. For all submitted tasks in Meta-task; Ti
1.1. For all resources; Rj
1.1.1. Cij = Eij + rj
2.While Meta-task not Empty
2.1 Find task Tk costs Geometric Average or nearest Greater than Geometric Average execution time.
2.2 Assign task Tk to resource Rj which gives minimum completion time (Slowest resource).
2.3. Remove task Tk from Meta-tasks set.
2.4. Update rj for selected Rj.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)

on 21st & 22nd April 2014, Organized by

Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

2.6. Update Cij for all j.

IV. EXPERIMENTAL RESULTS

We have used the Inspiral_100 , Montage_100 and Sipht_100 workflows to test our algorithm. Makespan and average utilization of the resource are used to evaluate the performance of the algorithm. The Makespan of a workflow is the time elapsed from its submission to the cloud until the completion of its last task. For the multi-workflow scheduling experiments, in which a number of concurrent instances of a workflow are submitted to the cloud, we consider the metric Total Makespan, which is the time elapsed from the submission of the instances until all the instances are completed.

Here we considered two resources R1 and R2 with the processing speed 100 and 400 MIPS respectively. The tool used to simulate the experiments is WorkflowSim. WorkflowSim is a workflow simulator to support large-scale scheduling, clustering and provisioning studies. It extends the existing CloudSim simulator [7] by providing a higher layer of workflow management. It also ignore system overheads and failures in simulating scientific workflows could cause significant inaccuracies in the predicted workflow runtime.

The Table 1 shows the comparison of makespan of various versions of Max-min algorithms. The Results shows that our proposed algorithms gives better results

TABLE 1
MAKE SPAN OF MAX-MIN ALGORITHMS

	Inspiral_100	Montage_100	Sipht_100
Max-min	44367.42	2223.26	40747.92
IMax-min	42907.12	2166.074	36733.53
IMax-min1	42830.92	2166.45	36588.05
IMax-min2	42390	2165.45	35980.04

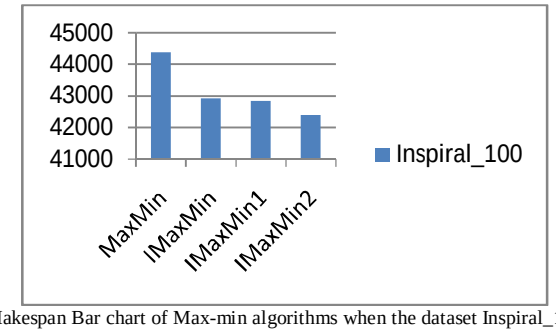

Fig 1 : Makespan Bar chart of Max-min algorithms when the dataset Inspiral_100 is used

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)

on 21st & 22nd April 2014, Organized by

Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

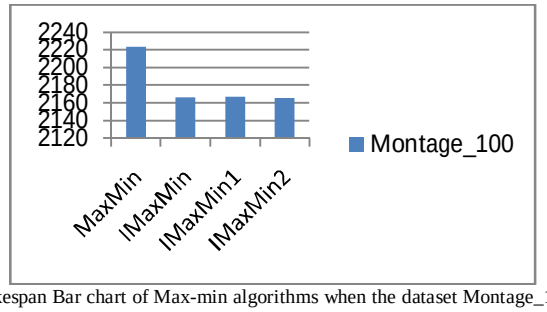

Fig 2: Makespan Bar chart of Max-min algorithms when the dataset Montage_100 is used

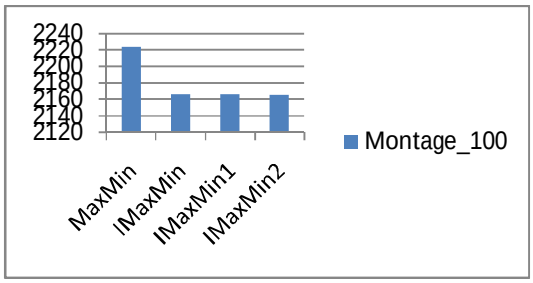

Fig 3: MakeSpan Bar chart of Max-min algorithms when dataset Sipht_100 is used

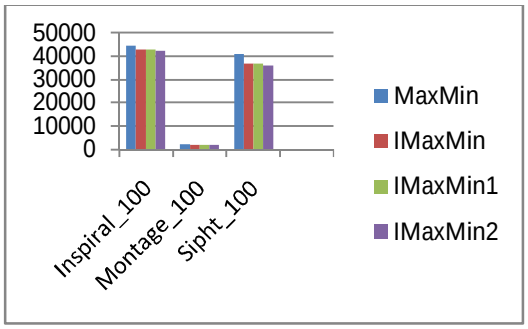

Fig 4:Bar chart of overall comparison of Make span results

The below table 2 shows the average utilization of various Max-min algorithms.

TABLE 2:
AVERAGE UTILIZATION OF RESOURCES

	Inspiral_100	Montage_100	Sipht_100
Max-min	44360.595	2223.26	40745.975
IMax-min	42762.16	2130.2	31798.71
IMax-min1	40874.96	2128.7	31316.93
IMax-min2	41635	2131.3	31456.92

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 2, May 2014

International Conference On Advances in Computer & Communication Engineering (ACCE - 2014)

on 21st & 22nd April 2014, Organized by

Department of CSE & ISE, Vemana Institute of Technology, Bengaluru, India

V. CONCLUSIONS AND FUTURE WORK

When we use Max-min algorithm always largest task will be assigned to the best available resource (fastest) and does not consider the completion time .In Improved Max-min completion time is considered and it assigns the largest task to the resource which gives minimum completion time. In cloud computing users any time can submit the job for execution and environment is dynamic. Cloud resource broker considering the dependency between the jobs , jobs are submitted batch wise dynamically[7] . Experimental results shows that instated of selecting the largest job every time selecting the average sized jobs results in better makespan and average utilization of resources.

The Study can be further extended by considering the nature of the jobs in the joblist – homogeneous or heterogeneous in size, the complexity still it can be improved. And also fault tolerance can be considered , because after a job is submitted to the resource , if the resource becomes unavailable it may affect the makespan and average utilization of the resource.

ACKNOWLEDGEMENT

The authors would like to thank the dedicated research group in the area of Cloud & Grid Computing , wireless networks at the Dept of Computer Science, Mangalore University, Mangalore, India, for many stimulating discussions for further improvement and future aspects of the Paper. Lastly but not least the author would like to thank everyone, including the anonymous reviewers.

REFERENCES

[1]A. Fox, R. Griffith et al., “Above the clouds: A berkeley view of cloud computing,” Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Tech. Rep. UCB/EECS, vol. 28, 2009.
[2] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” in Grid Computing Environments Workshop, 2008. GCE’08. IEEE, 2008, pp. 1–10.
[3] C. Germain-Renaud and O. F. Rana, “The convergence of clouds, grids, and autonomics,” Internet Computing, IEEE, vol. 13, no. 6, pp. 9–9, 2009.
[4] Saeed Parsa, Reza Entezari- Maleki. 2009. RASA: A New Grid Task Scheduling Algorithm. International Journal of Digital Content Technology and its Applications. Volume 3, Number 4.
[5] O M Elzeki, M Z Reshad and M A Elsoud. Article: Improved Max-Min Algorithm in Cloud Computing. International Journal of Computer Applications 50(12):22-27, July 2012. Published by Foundation of Computer Science, New York, USA.
[6] Upendra Bhoi1, Purvi N. RamanuJ”Enhanced Max-min Task Scheduling Algorithm in Cloud Computing” International Journal of Application or Innovation in Engineering &Managenet (IJAIEM)” April 2013 ISSN 2319 - 4847
[7] Weiwei Chen; Deelman, E., "WorkflowSim: A toolkit for simulating scientific workflows in distributed environments," E-Science (e-Science), 2012 IEEE 8th International Conference on , vol., no., pp.1,8, 8-12 Oct. 2012
[8] R. N. Calheiros, R. Ranjan,A. Beloglazov, C. A. F. De Rose, and R. Buyya. "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Software: Practice and Experience, 41(1): 23-50, Wiley, January 2011
[9] Chieu T.C., Mohindra A., Karve A.A., Segal A., “Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment ,” in IEEE International Conference on e-Business Engineering, Dec. 2009, pp. 281-286.
[10] Naidila Sadashiv, S. M Dilip Kumar, “Cluster, Grid and Cloud Computing: A Detailed Comparison,” The 6th International Conference on Computer Science & Education (ICCSE 2011) August 3-5, 2011. SuperStar Virgo, Singapore, pp. 477- 482.
[11] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer, Ivan Breskovic, “SLA-Aware Application Deployment and Resource Allocation in Clouds”, 35th IEEE Annual Computer Software and Application Conference Workshops, 2011, pp. 298-303.
[12] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, “Low level metrics to high level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments,” In High Performance Computing and Simulation Conference, pages 48 – 55, Caen, France, 2010.
[13] M. Maurer, I. Brandic, V. C. Emeakaroha, and S. Dustdar, “Towards knowledge management in self-adaptable clouds,” In 4th International Workshop of Software Engineering for Adaptive Service-Oriented Systems (SEASS’10) , Miami, Florida, USA, 2010.
[14] T. Hagras and J. Janecek, “A high performance, low complexity algorithm for compile-time task scheduling in heterogeneous systems,” Parallel Computing, vol. 31, no. 7, pp. 653–670, 2005.