Turnitin Originality Report

VIT paper by Geethu Pp

From Paper (CSE Papers)

Processed on 22-Jan-2015 12:03 PM SGT
ID: 497175309
Word Count: 2343

Similarity Index	Similarity by Source	
14%	Internet Sources:	7%
	Publications:	9%
	Student Papers:	7%

sources:

1 3% match (publications)
Kumar, Pardeep, and Amandeep Verma. "Scheduling using improved genetic algorithm in cloud computing for independent tasks", Proceedings of the International Conference on Advances in Computing Communications and Informatics - ICACCI 12 ICACCI 12, 2012.

2 2% match (student papers from 24-Dec-2014)
Submitted to ABV-Indian Institute of Information Technology and Management Gwalior on 2014-12-24

3 2% match (student papers from 18-Dec-2014)
Submitted to Malaviya National Institute of Technology on 2014-12-18

4 1% match (Internet from 24-Oct-2014)
http://www.researchgate.net/publication/245031122_Distributed_mixed_variant_differential_evolution_algorithms_for_unconstrai

5 1% match (student papers from 06-Oct-2014)
Submitted to Malaviya National Institute of Technology on 2014-10-06

6 1% match (Internet from 16-Mar-2014)
http://ijcsi.org/papers/IJCSI-9-1-1-238-246.pdf

7 1% match (Internet from 04-Mar-2009)
http://halcyon.usc.edu/~yangyu/data/HCW02.pdf

8 1% match (publications)
Maheswaran, M.. "Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems", Journal of Parallel and Distributed Computing, 199911

9 < 1% match (publications)
Y. Inoguchi. "Practical Task Flow Scheduling for High Throughput Computational Grid", 2006 International Conference on Parallel Processing Workshops (ICPPW 06), 2006

10 < 1% match (Internet from 12-Sep-2014)
http://ijsret.org/pdf/120559.pdf

11 < 1% match (Internet from 22-Aug-2014)
http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-5-1672-1676.pdf

12 < 1% match (publications)
Ghosh, Tarun Kumar, Rajmohan Goswami, Sumit Bera, and Subhabrata Barman. "Load balanced static grid scheduling using Max-Min heuristic", 2012 2nd IEEE International Conference on Parallel Distributed and Grid Computing, 2012.

13 < 1% match (Internet from 22-Aug-2014)
http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-4-1323-1329.pdf

14 < 1% match (Internet from 01-Mar-2008)
http://www.engr.colostate.edu/~aam/pdf/journals/51.pdf

15 < 1% match (Internet from 08-Apr-2014)
http://ceit.aut.ac.ir/~meybodi/paper/Shojafar-Pooranian-Abawajy-Meybodi-JCSE-2013=V7N1-04.pdf

16 < 1% match (Internet from 06-Nov-2014)
http://arxiv.org/ftp/arxiv/papers/1403/1403.2914.pdf

17 < 1% match (publications)
Turgun T.. "NDA-MM: A New Adaptive Task Scheduling Algorithm Based on the Non-dedicated Constraint Grid", Sixth International Conference on Grid and Cooperative Computing (GCC 2007), 08/2007

18 < 1% match (publications)
Juefu Liu. "The research of load imbalance based on Min-Min in grid", 2010 International Conference On Computer Design and Applications, 06/2010

paper text:

2**Available online at www.sciencedirect.com ScienceDirect Procedia Computer Science00 (2015) 000–000 www.elsevier.com/locate/procedia** 2nd **International** Symposium **on** Big Data **and**

Cloud Computing (ISBCC'15) An in-depth analysis and

13**study of Load balancing** techniques **in** the **cloud computing environment.**

Geethu Gopinath P P(1), Shriram K Vasudevan(2)

4**Department of Computer Science and Engineering** (1) **Amrita School of Engineering, Amrita Vishwa Vidyapeetham (University),** Ettimadai, **Coimbatore —**

641112, geetkrish90@gmail.com Asst professor (senior)(2)

4**Department of Computer Science and Engineering, Amrita School of Engineering, Amrita Vishwa Vidyapeetham (University),** Ettimadai, **Coimbatore —**

641112, Kv_shriram@cb.amrita.edu ph: 08939918562 Abstract In the cloud computing paradigm,

6**load balancing is one of the challenges,** With Tremendous increase **in**

the users and their demand of different services on the cloud computing platform, fruitful or efficient usage of resources in the cloud environment became a critical concern. Load balancing is playing a vital role in maintaining the rhythm of Cloud computing. The performance metrics of load balancing algorithms in cloud are response time and waiting time. In this paper we mainly focus on two load balancing algorithms in cloud, Min-Min and Max-Min

2**algorithm. © 2015 The Authors. Published by Elsevier B.V. Peer-review under responsibility of** scientific **committee of** 2nd **International** Symposium **on** Big Data **and**

Cloud Computing (ISBCC'15). Keywords: Min-Min, Max-Min, Response time, Load Balancing, Batch mode scheduling, Immediate mode scheduling, Make span, Minimum completion time, Minimum execution time

5**1877-0509© 2015 The Authors. Published by Elsevier B.V. Peer-review under responsibility of scientific committee of** 2nd **International** Symposium **on** Big Data **and**

Cloud Computing (ISBCC'15). 1. Introduction Load balancing [1] is a method that distributes the workload among diverse nodes in the given environment such that it ensures no node in the system is over loaded or sits idle for any instant of time. An efficient load balancing algorithm will make sure that every node in the system does more or less same volume of work. The responsibility of load balancing algorithm is that to map the jobs which are set forth to the cloud domain to the unoccupied resources so that the overall available response time is improved as well as it provides efficient resource utilization. Balancing the load became one of the crucial concerns in cloud computing since we cannot predict the number of requests that are issued at each second in cloud environment. The unpredictability is due to the ever changing behaviour of the cloud. The main focus of load balancing in the cloud domain is in allocating the load dynamically among the nodes in order to satisfy the user requirements and to provide maximum resource utilization by assorting the overall available load to distinct nodes. Fig 1. Diagram for load balancing 2. Demand

> 6**of Load Balancing in Cloud** Environment **Load balancing** [1] **is a** method such **that** it assigns **the**

workload equally among all the available nodes which are present in the system. Higher user satisfaction is the motto behind load balancing. As the number of user's as well as their demands are increasing day by day, the clouds should provide the services to the customers with their at most satisfaction. An appropriate or an ideal load balancing algorithm help in making use of the available resources most favourably, thereby ensuring no node is over loaded or under loaded. Load balancing enables scalability, avoids bottlenecks and also reduces time taken to give the respond. Many load balancing algorithm [2] have been designed in order to schedule the load among various machines. But so far there is no such ideal load balancing algorithm has been developed which will allocate the load evenly across the system. It has been proved that allocating the tasks evenly across the system is considered to be an NP complete problem [7]. 3 3.

> 10**Load balancing in** a **cloud computing** environment **Load balancing** algorithms **in cloud**

environment are manly classified in to two categories which are ?

> 9**Immediate mode scheduling ? Batch mode scheduling Immediate mode scheduling** schedules **the** task on **to the**

resources based on its arrival. In immediate mode scheduling the algorithms used are MET which is expanded as minimum execution time and minimum completion time which is abbreviated to MCT. In MCT algorithm the task which is assumed to have the minimum completion time will get allocated to the analogous node. In Batch mode scheduling tasks are collected based upon their arrival in a Meta task set and they are mapped at prescheduled times to their corresponding machines.

> 18**Min-Min and Max- Min** belongs to **the** Batch mode **scheduling**

category. 3.1 Min-Min load balancing algorithm The Algorithm take up with a task set which are initially not assigned to any of the nodes. Initially

> 15**the minimum completion time is calculated for** all **the**

available nodes. Once this calculation gets completed the task having the

> 8**completion time** minimum **is** chosen **and assigned to the** respective node. **The**

execution time of all other tasks which are currently available in that machine is updated and the task gets discarded from the available task set. The routine is done time after time until all the tasks have been assigned to the equivalent machines. The algorithm works better when the situation is like where the small tasks are greater in number of than the large tasks. The algorithm has a disadvantage that it leads to starvation. Min-Min is a simple and fast algorithm capable of providing improved performance. Min-Min schedules the ideal tasks at first which results in best schedules and improve the overall make span. Assigning small task first is its drawback. Thus, smaller tasks will get executed first, while the larger tasks keeps on in the waiting stage ,which will finally results in poor machine use. Min-Min exhibits minimum completion time for jobs which are unassigned (similar to MCT), and later allocating the jobs with minimum completion time (hence min-min) to a node that is capable of handling it. Architectural description of Min-Min algorithm is shown below in Fig. 2 Fig. 2 Architectural description of min-min algorithm.

13.2 Max Min load balancing algorithm The max-min algorithm is much the same as to min-min algorithm.

At first for all the available tasks are submitted to the system and minimum completion time for all of them are calculated, then among these tasks the one which is having the

16completion time, maximum is chosen and that is allocated to

the corresponding machine. This algorithm outperform than Min-Min algorithm where when short tasks are in high numbers when compared to that of long ones. For e.g. if in a task set only a single long task is presented then

12,Max Min algorithm runs short tasks concurrently along with long task.

The make span focus on how much small tasks will get executed concurrently with the large ones.

8Max-Min is almost identical to Min-Min,

except it selects

11the task having the maximum completion time and allocates to the

corresponding machine. The algorithm suffers from starvation where the tasks having the maximum completion time will get executed first while leaving behind the tasks having the minimum completion time. Architectural description of Max Min algorithm is presented below in Fig. 3 Fig .3 Architectural description of max-min algorithm. 5 4. Simulation overview In this section, we are providing an experimental result which shows the comparison between the

14Min-Min and Max-Min algorithms. For the implementation of both the

algorithms

1we have used CloudSim [8], a simulator for checking the performance of

the two algorithms. “CloudSim [9]

1is an extensible simulation toolkit that enables modelling and simulation of Cloud computing systems and application provisioning environments”. In cloudsim

tasks are considered to be the cloudlets and nodes are taken into account as virtual machines. Using cloudsim the performance of these algorithm is observed in three

1cases: in first case, we have set the number of

nodes to be five which is kept as constant for all the cases and changed the number of tasks to 25.For the second case we have varied the tasks to 50 later in the third case it is changed to 100.Experiments conducted with twenty five, fifty and hundred tasks are assigned to Cloud with 5 resources. The resources are located to one data centre. Table 1 shows the simulation results of both the algorithms. It shows the make span of

17Min-Min and Max-Min algorithm for three cases Table 1. Make span of two algorithms Number of tasks 25 tasks 50 tasks 100 tasks Min Min 159 230 510 Max Min 120 200 450 From the above table it is noted that make span get reduced for Max-Min compared with that

1**of Min-Min, Max-Min** outperforms than **the**

other and in the later run

1**the assignment of** tasks **to the** resources get changed. **i.e. if**

we are using these two techniques the allocation of tasks to the machine will not be the same it will get changed. Depending upon the type of load balancing algorithm we choose the tasks are allocated to the respective nodes. Fig. 4 Results of comparison of Max-Min & Min-Min The performance of the three cases in accordance with the computed values are shown in chart above as shown in Fig. 4, in which y-axis shows the make span and x-axis shows the two algorithms. From the above simulation results we came into a conclusion that Max-Min achieves better performance than Min-Min with respect to the make span. For an efficient load balancing algorithm it always tries to reduce the make span 5. Conclusion We have done an extensive study through the implementation of the two load balancing algorithms namely Max-Min and Min-Min based on our selected cloud environment. The result of our evaluation shows that the

7**Max-Min performs better than Min-Min** in terms **of**

make span. But there are other works of load balancing in cloud environment which shows that Min-Min outperform the Max-Min algorithm. Both algorithms have got their own pros and cons, where depending upon the cloud environment one outperforms the other. If the number of lighter tasks outnumbers the heavier tasks then

7**Max-Min performs** well **better than** the **Min-Min** in terms **of** resource utilisation **and**

make span on the contrary if there are many heavier tasks it results in Min-Min to perform better than Max-Min. So we came in to a conclusion that the performance of load balancing in cloud doesn't depend upon any algorithms but it is purely based on the cloud environment we choose. 6. References 1. N. Ajith Singh, M. Hemalatha, "An approach on semi distributed load balancing algorithm for cloud computing systems" International Journal of Computer Applications Vol-56 No.12 2012 2. Shanti Swaroop moharana, Rajadeepan d. Ramesh & Digamber Powar ," Analysis of load balancers in cloud computing" International Journal of Computer Science and Engineering (IJCSE)ISSN 2278-9960 Vol. 2, Issue 2, May 2013, 101-108. 3. Gaurav R. et al. "Comparative Analysis of Load Balancing Algorithms in Cloud Computing." International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1, No. 3, pp.120-124, May 2012. 4. A.Khiyaita, M.Zbakh, H. El Bakkali & Dafir El Kettani ,"Load Balancing Cloud Computing : State of Art" IEEE Network Security and Systems (JNS2), 2012 National Days of , 978-1-4673-1050-5. 5. Ektemal Al-Rayis, Heba Kurdi," Performance Analysis of Load Balancing Architectures in Cloud Computing" IEEE Modelling Symposium (EMS), 2013 European, 20-22 Nov. 2013, 978-1-4799-2577-3. 6. Shridhar G.Damanal and G. Ram Mahana Reddy," Optimal Load Balancing in Cloud Computing By Efficient Utilization of Virtual Machines", IEEE 978-1-4799-3635-9/14 7. D. Fernández-Baca: Allocating modules to processors in a distributed system, IEEE Transactions on Software Engineering, Vol. 15, No. 11, pp. 1427-1436 (1989). 8. Rodrigo N. Calheiros, Rajiv Ranjan,Anton Beloglazov and Rajkumar Buyya,Cesar A. F. De Rose," CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms" Published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995 9. Dr. Rahul Malhotra & Prince Jain," Study and Comparison of CloudSim Simulators in the Cloud Computing", The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), Vol. 1, No. 4, September-October 2013. 10. Haozheng Ren, Yihua Lan, Chao Yin," The Load Balancing Algorithm in Cloud Computing Environment", 2012 2nd International Conference on Computer Science and Network Technology,IEEE 978-1-4673-2964-4/12. 11. Dr. Rahul Malhotra, Prince Jain," Study and Comparison of Various Cloud SimulatorsAvailable in the Cloud Computing" International Journal of Advanced Research in Computer Science and Software Engineering,Volume 3, Issue 9, September 2013 ISSN: 2277 128X 12. Tejinder Sharma, Vijay Kumar Banga," Efficient and Enhanced Algorithm in Cloud Computing", International Journal of Soft Computing and Engineering (IJSCE)ISSN: 2231-2307, Volume-3, Issue-1, March 2013. 13. Sujit Tilak, and Patil.D, "A Survey of Various Scheduling Algorithms in Cloud Environment",International Journal of Engineering Inventions,September 2012,pp.36-39. 14. 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. 15. Bhoi U, and Ramanuj P. N. "Enhanced Max-min Task Scheduling Algorithm in Cloud

Computing”.International journal of Application or innovation in Engineering & Management(IJAIEM), Volume 2, Issue 4, April 2013 16. O. M. Elzeki, M. Z. Reshad, M. A. Elsoud, “Improved Max-Min Algorithm in Cloud Computing ", International Journal of Computer Applications (0975 – 8887) Volume 50 –No.12, July 2012. 17. Rima P. Lingawar , Milind V Srode ,Mangesh M. Ghonge,” Survey on Load-Balancing Techniques in Cloud Computing”, International Journal of Advent Research in Computer & Electronics, Vol.1, No.3, May 2014E-ISSN: 2348-5523. 18. Nidhi Jain Kansal, Inderveer Chana,” Cloud Load Balancing Techniques : A Step Towards Green Computing”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012. 19. T. Kokilavani,” Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing” International Journal of Computer Applications (0975 – 8887) Volume 20– No.2, April 2011.

3**2 Author name / Procedia Computer Science00** (2015) **000–000 Author name / Procedia Computer Science00** (2015) **000–000 4 Author name / Procedia Computer Science00** (2015) **000–000 Author name / Procedia Computer Science00** (2015) **000–000 6 Author name / Procedia Computer Science00** (2015) **000–000**