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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Deadline-Aware Sufferage Based task Scheduling in Cloud Computing

Alka Vohra, Suchitra

M.Tech Student, Dept. of CSE, Galaxy Global Group of Institution, Kurukshetra University, Dinarpur, Ambala, India

Assistant Professor, Dept. of CSE, Galaxy Global Group of Institution, Kurukshetra University, Dinarpur, Ambala, India

ABSTRACT: Cloud computing is the use of computing resources that are delivered as a service over a network. It supplies a high performance computing based on protocols which allows shared computation and storage over long distances. In cloud computing, many tasks need to execute at a time by the available resources in order to achieve better performance, minimum completion time, resource utilization etc. Because of these different factors, we need to design, develop, and propose a scheduling algorithm for the proper allocation of tasks to the resources. In this dissertation, a modification of DA-Sufferage (Deadline Aware) algorithm is proposed. This algorithm based on concept of Sufferage strategy. A DA-Sufferage (Deadline Aware) algorithm is developed to outperform scheduling process. So the scheduling tasks within cloud environment using DA-Sufferage (Deadline Aware) can achieve lower makespan and higher resource utilization.

KEYWORDS: Cloud computing, Min-Min, Max-Min, DA-Sufferage, Makespan, Average Resource Utilization

I. INTRODUCTION

Cloud computing is a new kind of model, is coming. This word is a new word that appears at the fourth season, 2007. The use of Internet and new technologies nowadays, for business and for the current users, is already part of everyday life. Any information is available anywhere in the world at any time. That was not possible few years ago. Nowadays it have arisen a lot of possibilities of access to public and private information like internet speed access or the deployment of mobile dispositive that allow the connection to Internet from almost everywhere.

Cloud computing refers to the delivery of computing resources over the Internet. Instead of keeping data on your own hard drive or updating applications for your needs, you use a service over the Internet, at another location, to store your information or use its applications. Doing so may give rise to certain privacy implications. Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations.

Cloud computing provide digital service delivery over the internet by various applications that are that are accomplished by computer system in distributed datacentres. Cloud coping getting more advance now days. Cloud service providers are aimed to provide services using large scale cloud environment with cost effectiveness. Also there are few large scaled applications like social-networking-commerce etc. these applications can benefit to minimize the cost using cloud computing. It provide infrastructure, platform and software which are made available as subscription-based services in a pay- as- you go model to customers.

II. RELATED WORK

In [1] author used a novel QoS guided task scheduling algorithm for Grid computing is presented. The proposed novel calculation depends on a general adaptive scheduling heuristics that incorporates QoS direction. The algorithm is assessed inside a simulated Grid condition. The exploratory outcomes demonstrate that the new QoS guided Min-Min

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

heuristic can prompt huge execution pick up for an assortment of utilizations. In [2] author used RASA algorithm. RASA is improve version of Max-Min algorithm. In [3] author used standard genetic algorithm that cover all the strategies of Min-Min and Max-Min. This system can be adjusted to the cloud processing frameworks for better scheduling of tasks to assets so that the clients' tasks can be finished in as least time as would be prudent. In [4] author used different task scheduling algorithm. These job scheduling algorithms are investigated in view of the makespan, flow time, and asset usage and fruition time. Job scheduling is the preeminent stride in lattice registering where the clients' jobs are scheduled to various machines.In [5] author used Location based Minimum Migration in Cloud (LBMMC). This algorithm monitors all the virtual machine in all cloud location.It also identifies the state of virtual machines whether it is sleep or idle state it also checks the number of tasks able to run or check number of tasks running. In [6] author used Min-Min Ant Colony (MMAC) algorithm. This algorithm that lessens the makespan and augment the asset usage utilizing the components of both min-min calculation and ant colony enhancement. This is two-stage algorithm. The exploratory outcomes gotten by applying the proposed algorithm for different issues appears that it beats the current scheduling algorithms. In [7] author used improved Max-Min algorithm. In the approved new Max-Min algorithm largest feature is selected and assigned to resource which takes minimum completion time.

III. PROPOSED ALGORITHM

The Proposed approach consider two parameters (1) Task length (2) User Priority. The algorithm is based on deadline aware rank system. Each task is assigned a task based on their task length and priority. In the actual scheduling of task, these rank will be considered.

A. Task Length Rank

The cloud system executes tasks having different length. If the tasks are arranged based on the increasing order of length, tasks having shorter length will reside at the beginning of the array and the task having highest length will be present at the last. For the scheduling purpose, the algorithm should take tasks from both front and back, giving it a bit more stability. The task priority rank system based on task length will work as follows: The first step is involved in finding the length of each task (Tleni). The second step is calculating the average of tasks length (lenavg). Third step begins with the calculation of difference in length with respect to lenavg. Let task set be T1, T2, T3.etc. Here equation 2 is used for finding the difference in length with the average length. This data is useful when tasks are arranged in an array in an increasing order of task length. The proposed algorithm neither takes task with larger length nor task with lower length. It takes each tasks from the middle.

TLDi = lenavg–Tleni (2)

Where TLDi is the task length difference of task i. It is computed by taking the absolute difference of task length for the ith task and average value. After finding the difference in task lengths of each task, ranks areassigned to each task. In this algorithm there are 5 ranks and these ranks are given to each tasks for different conditions. Before these steps, 4 different values are found from the length array. These 4 values forms the condition for assigning the ranks. We can't simply choose 4 values. These values should be in a range of task length. The computations are given below.

value_1=high-len/5 (3)

value_2=high-len/4 (4)

value_3= value_2+ value_1 (5)

value_4= value_3+ value_2 (6)

Where high-len is the highest value of task length. This can be found by Pseudo code which is mentioned below.

For all submitted tasks in the set; Ti

| TLDi = lenavg - Tleni|

If TLDi≤ value_1

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Then rank =5

Else

If value_1 <TLDi≤ value_2

Then rank =4

Else

If value_2 <TLDi≤ value_3

Then rank =3

Else

If value_3 <TLDi≤ value_4

Then rank =2

Else

value_4 >TLDi

Then rank =1

End For

This algorithm adds ranks based on the task length. After this step each task will associated with a rank (Rank-Lengthi)

B. Task priority rank based on deadline

Task priority is also important for scheduling tasks. Each task may have different priorities, which are represented as values assigned to each task and the value can be the same for more than one task. Priority of a task can be computed by equation (7). The scheduling algorithm based on task priority has the problem of treating tasks with similar priority. In the proposed approach this does not arise as a problem because even though we are giving credits to each tasks based on their priority, the final scheduling will be based on total credit which is based on task length and its priority. Here in this algorithm different tasks are assigned different credits. Credit value is generated based on the priority which is assigned to the each task. In the proposed approach priority average is not important. Suppose there are 10tasks, then there will be 10 different rank. There will be 20 rank when dealing with 20 tasks. The fact is that these rank are not set by default. The credit value will change based on the priority that is assigned by the user.

IV. PSEUDO CODE

Step 1: Assign task priority (max user pay) accord to prediction model:
Step 2: Pi= k*Li/Di (Li-length of task i, Di-deadline of task i, priority of task i) (7)
Step 3: For all submitted tasks in the set; Ti
Step 4: Find out task with maxPi=highest priority (Priority pi)
Step 5: Choose division-part = (10^numdigits (maxPi))
Step 6: For each task with priority Tpri
Step 7:findPri_frac(i)=Tpri / maxPi
Step 8: Set credit as Pri_frac

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

The primary step in the above algorithm is finding the highest priority number. Second step in the algorithm is choosing the division factor for finding Pri_frac for each task. This can be calculated by dividing priority value of each task with division factor of corresponding task. Finally this value (Pri_frac) will be assigned to each task as rank credit.

Total_Crediti= Credit_Lengthi*Credit_ranki (8)

The two credits are calculated separately. The final step in the algorithm is to find out the total credit based on task length and task priority. The Total credit is calculated by using the equation shown above. In the above equation Credit_Lengthi is the credit based on task length. Credit_ranki is the credit based on rank. Finally task having highest rank will be scheduled first.

V. Simulation Results

This method contain 100 tasks and 10 Virtual Machines (Vm).In this method we randomly generate the tasks. The length of each cloudlets re ranges from 1000 to 10000. The output is as show below:

Table 1: Result of 100 task and 10 Vm

Algo	Makespan	Succeed task	Fail task	Provider Profit	User Loss	Average RU
MIN-MIN	53.1	87	13	30754.06	3309.88	0.52
MAX-MIN	44.1	91	9	31255.21	2999.02	0.61
DA-Sufferage	**24.54**	**100**	**0**	**34784.61**	**0**	**0.92**

Makespan

Makespan is a measure of the throughput of the heterogeneous cloud system. It can be calculated as the following relation:
Makespan=max (rtj)
Where rtj denotes the ready time of each resource after scheduled. The less the makespan of a scheduling algorithm the better it works.

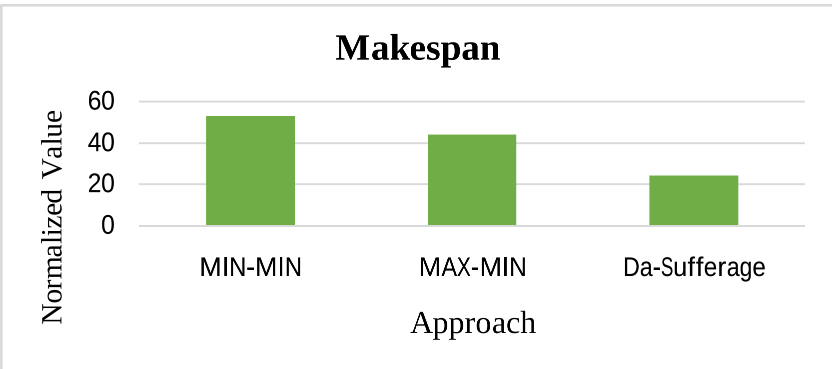

Fig 1: Makespan

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Makespan of DA-sufferage is better as compared to Min-Min and Max-Min. We have to reduce the makespan.

Task Profit

Task Profit is defined as number of cloudlets which have completed successfully before they meet the deadline i.e. number of cloudlets which meet the deadline before it decided.

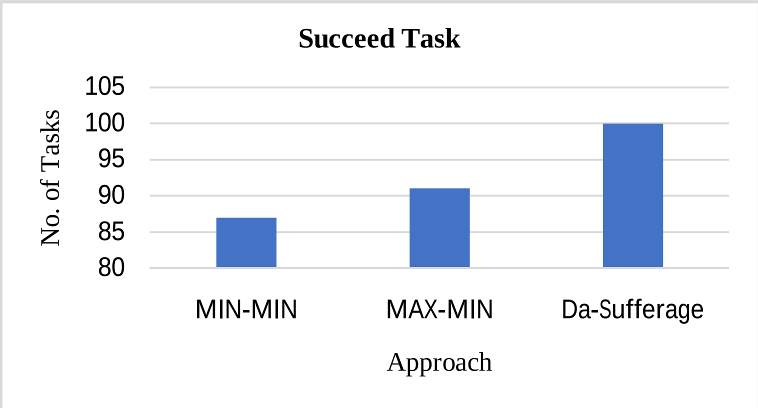

Fig 2: Succeed task

We can easily see the difference that the succeed task in DA-Sufferage is better as compared to other two algorithm Min-Min and Max-Min.

Task Penalty

Task Penalty is defined as the number of cloudlets misses their deadline i.e. number of tasks which are completed successfully within the deadline.

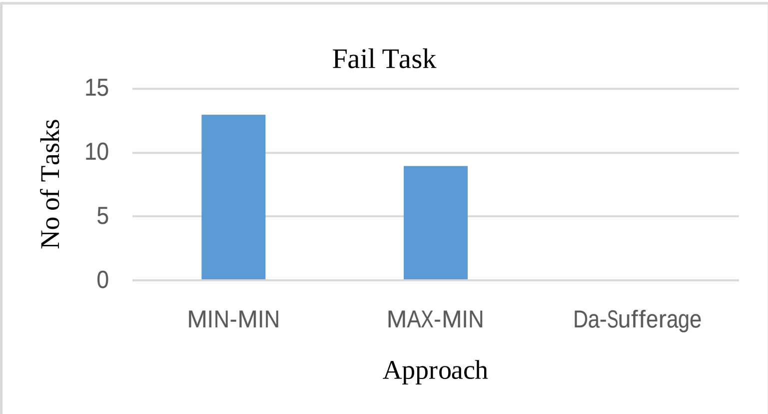

Fig 3 Task Penalty

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Average Resource Utilization

ARUR is calculated through the following relation:

ARUR= mean (rtj)/Makespan*100%

Where ARUR is in the range 0 to 1. Scheduling algorithm will have better performance if ARUR is close to 1.

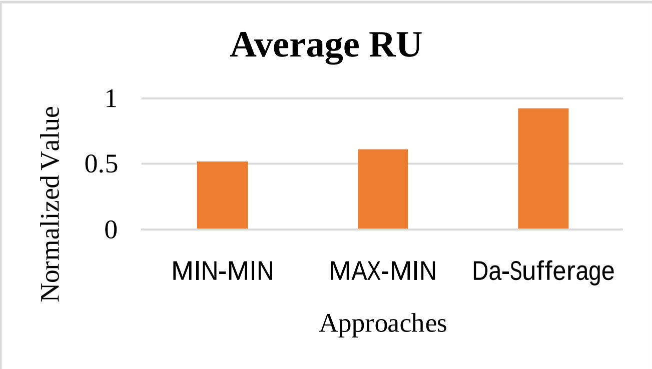

Fig 4: Average Resource Utilization

VI. CONCLUSION AND FUTURE WORK

When we compare Min-Min, Max-Min and DA-Sufferage that we have proposed, we can easily notice that DA-Suffrage performance better than these two in-Min and Max-Min algorithm. Makespan of DA-Sufferage is better and Average resource utilization is improved of this algorithm. This has become possible by introducing the concept of deadline. The concept of deadline is also used in Min-Min but DA-Sufferage perform well using deadline concept. Whenever it is compared with cost and deadline based Min-Min it shows that makespan of proposed algorithm is better. Min-min and Max-min algorithms are most popular algorithm used in small scale distributed systems. When the number of small tasks is higher than number of large tasks in a meta-task, in that case Sufferage works well.

In this proposed algorithm we have assign a rank for each task to sort them and choose DA-Sufferage schedule for task schedule. This model give more advantages than existing work in terms of task arrangement based on deadline using composite credit based on task length and deadline of the task. We perform analysis of the proposed model with existing technique to predict the improvement regard make-span, net profit, task penalty and Average resource utilization. In future, we can try to optimize the schedule using optimization algorithms like GA, PSO etc.

REFERENCES

[1] HE XiaoShan , SUN XianHe , Gregor von Laszewski ," Qos Guided Min-Min Heuristic For Task Scheduling" in J. Comput. Sci. & Technol. July 2003, Vol.18, No.4, pp.442-451

[2] Elzeki, O. M., M. Z. Reshad, and M. A. Elsoud. "Improved Max-Min Algorithm in Cloud Computing", International Journal of Computer Applications, Volume 50, Issue 12, Pages 22-27, 2012.

[3] Pardeep Kumar, Amandeep Verma , " Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm" in International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X Volume 2, Issue 5, May 2012.

[4] G. Jaspher W. Kathrine, MansoorIlaghi U," Job Scheduling Algorithms in Grid Computing" in International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 1 Issue 7, September – 2012

[5] Ms.K.Sathya, Dr. S.Rajalakshmi, "Deadline Based Task Scheduling in Cloud with Effective Provisioning Cost using LBMMC Algorithm", International Journal On Engineering Technology and Sciences – IJETS™ ISSN (P): 2349-3968, ISSN (O): 2349-3976 Volume 1 Issue 7, November 2014

[6] Davinder Kaur, Sarpreet Singh , "An Efficient Job Scheduling Algorithm using Min-Min and Ant Colony Concept for Grid Computing" in International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 03 Issue 07 July, 2014 Page No. 6943-6949.

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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

[7] Santhosh B and Manjaiah D H, "A hybrid AvgTask-Min and Max-Min algorithm for scheduling tasks in cloud computing," 2015 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), Kumaracoil, 2015, pp. 325-328.

[8] L. Zuo; L. Shu; S. Dong; Y. Chen; L. Yan, "A Multi-objective Hybrid Cloud Resource Scheduling Method Based on Deadline and Cost Constraints," in IEEE Access , vol.PP, no.99, pp.1-1

[9] Rajwinder Kaur and Pawan Luthra , “Load Balancing in Cloud Computing” in ACEEE, Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC

[10] N. Patil and D. Aeloor, "A review - different scheduling algorithms in cloud computing environment," 2017 11th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, 2017, pp. 182-185