International Journal of Research in Advent Technology, Vol.7, No.4, April 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

Priority Based Resource Allocation For Meta-Task Scheduling In Cloud Computing

S. Kavitha[1], Dr. D. I. George Amalarethinam[2]
[1] *Assistant Professor, Department of Computer Science, Saradha Gangadharan College, Pondicherry, India*
kavidharan2000@gmail.com
[2]*Bursar, Director MCA, Department of Computer Science, Jamal Mohamend College Trichy, India*
di_george@ymail.com

Abstract - Cloud computing environment assures to provide on demand computing services through internet on pay-as-you-use basis. Even though there are several issues existing, scheduling is the main issue in cloud. Scheduling of tasks is difficult because of sole elements in cloud computing such as resource heterogeneity, task heterogeneity and user priority. This paper proposes a new Priority based Resource Allocation (PRA) for meta-task scheduling in cloud. This algorithm thinks about priority given to meta-task by the users. The instances of high speed resource will be used for scheduling the meta-task set with user priority and heterogeneous resources are allocated for non-priority meta-task set. The PRA algorithm minimizes the makespan and cost, and increases the utilization of resources over the existing algorithm.
Index Terms - Meta-task set, Min-Min algorithm, priority, Resource allocation, Cost, Makespan.

1. INTRODUCTION

Computing in the Cloud is defined by a collection of resources that are used to compute and communicate sited in disseminated data centres that is distributed among various clients [1]. Scheduling is the main challenge in cloud environment. Several parameters like makespan, resource utilization, fault tolerance, load balancing, energy efficiency, cost, deadline, priority are used in task scheduling [2][3].

Scheduling of tasks is vital process in cloud environment. The process of assigning meta-tasks to the computing resources comes under NP complete [4]. Scheduling algorithms can be categorized based on the constraints as batch and immediate scheduling, non-pre-emptive and pre-emptive scheduling, dynamic and static scheduling, priority and non-priority scheduling [5].

The key goal is scheduling user-priority meta-task set in optimized way in cloud environment.

2. RELATED WORK

The priority is classified as execution-level priority, task level priority and user level priority. User level priority for meta-task set in cloud environment is considered in this paper.

The background work of this proposed algorithm is as follows:

Huankai Chen et al [6] have used Min-Min in first step and rescheduled the smallest task from heaviest resource to the resource which produced least completion time in second step.

Bhawna Taneja [7] has studied the Most Fit and Min-Min Priority task scheduling algorithms. The basic concept in Most Fit task scheduling policy is to assign the tasks to the most appropriate resource which gives minimum completion time; Whereas Min-Min Scheduling strategy allocates smallest tasks to most capable machines, which give less time to finish the task.

Naseem A.AL-Sammarraie et al [8] designed PAC. In PAC three priority levels are used and hence, the scheduler had three queues. After assigning the task's priority, it is sent to the suitable queue and specified scheduling algorithm is used for scheduling.

Er. Rajeev Mangla et al [9]. Recovery policy is used in RPA-LBIMM, to the scheduler in cloud for rescheduling the tasks during execution when resource fails.

Bhavisha Kanani et al [10]. Execution level priority is considered in this article. Priority is based on cost of CPU, memory and bandwidth. This strategy has three priority queues such as high, low and normal. Min-Min algorithm is used to schedule highest priority, medium priority and finally low priority respectively.

Pankajdeep Kaur et al [11]. Scheduling based on Priority had two policies. (1) Tasks with high priority is scheduled earlier than low priority tasks. (2) to decrease the resource cost by completing the task as early as possible.

Mokhtar A. et. al [12] have designed a new strategy called Scheduling Cost Approach (SCA) which calculates the cost of CPU, RAM, bandwidth and storage available. This is based on user priority which satisfies user budget.

Hitendra Pal et. al [13] have proposed Deadline Aware Modified Genetic algorithm. In this priorities and burst time of jobs are considered. Based on fitness value DAMGA improves the results.

Naoufal Er-raji et. al [14] have addressed the priority task scheduling problem. It considered the parameters such as tasks deadline, tasks age and task length over distributed data-centre in cloud. High priority is given to the task with minimum deadline.

Based on this review, the proposed algorithm considered user-priority during meta-task scheduling with guarantee that the users who pay high can enjoy better service.

3. SCHEDULING STRATEGY OF PRA ALGORITHM

In this paper, the proposed Priority based Resource Allocation Algorithm (PRA) been developed under a set of assumptions:

- The meta-task set in cloud environment is a collection of inseparable tasks.
- Calculation of expected task execution times on each Resource are done in advance
- Static scheduling is carried out.
- Priority is generated randomly.

In proposed algorithm, PRA, priority given by user is considered. Users who ready to pay high must have the benefit of good service. The highly paid user's tasks have priority. Remaining are non-priority tasks. In cloud environment, meta-task set is separated into two sets as priority tasks set and non-priority tasks. Initially priority task set are to be allocated to the instances of fastest resource by applying Rescheduling Enhanced Min-Min (REMM) algorithm. REMM algorithm follows 2 steps. In the first step, Enhanced Min-Min algorithm is used [15] and rescheduling the tasks based on the makespan is done in second step. Then non-priority tasks are allocated to the resources in heterogeneous cloud environment by applying REMM algorithm. PRA algorithm is shown in Figure-1.

1. Start
2. For all tasks in meta task set
3. For all resources m_j
4. Calculate $CT_{ij}=ET_{ij} + r_j$
5. Assign the user priority to the tasks
6. While all tasks are allocated
7. Design ETC-1 with Priority tasks
8. Design ETC-2 with Non-Priority tasks
9. Schedule the tasks in ETC-1 to the instances of fastest resource by REMM
10. Schedule tasks in ETC-2 to the resources in heterogeneous cloud environment by REMM
11. Stop

Fig. 1. Proposed Algorithm (PRA)

4. SIMULATION RESULTS IN CLOUD SIM

Cloud Sim: a good, widespread and extensible simulation tool which permits flawless representation, simulation, and testing of application services. The key benefits are: (i) time efficacy: it takes minimum attempt and execute time taken by applications in cloud. (ii) elasticity and pertinent: researchers may design and examine the efficiency of applications in cloud environment (iii) it has an autonomous platform for designing Clouds, service brokers, and allocation policies (iv) it has a virtualization engine which helps in the formation and organization of numerous, autonomous and virtualized services etc.,[16]. These convincing characteristics of Cloud Sim

must hustle up the progress of novel algorithms.Cloud parameters used in Cloudsim are as follows:

Table 1. CloudSim parameters

Parameters	**Values**
Total number of Tasks	10-1000
Total number of Resources	3-50
MIPS	1000-48000
VM Memory	512-4096

The user priority is randomly assigned when ETC is generated. The number of priority tasks, number of non-priority tasks, and number of resources for priority and non-priority tasks are shown in Table 2. The number of resources taken for scheduling is based on the following Eq. (1)

$$r = round\sqrt{n}) \quad (1)$$

Table 2. Makespan of different meta-task set in Cloud Sim

No of tasks			No. of resources		Makespan (in secs)	
TT	**PT**	**NPT**	**PT**	**NPT**	**MM**	**PRA**
10	7	3	3	2	100.66	71.01
25	6	19	3	4	98.17	88.27
50	18	32	4	6	224.9	203.13
100	41	59	6	8	209.14	187.34
250	59	191	8	14	306.83	287.85
500	461	39	21	6	620.71	556.0
750	729	21	27	4	331.72	309.48
1000	691	309	26	17	1568.0.1	1497.27

The makespan of Min-Min(MM) and proposed algorithm(PRA) are shown in Table 2. Figure 2. shows the performance of MM and PRA based on makespan. Table 3 and figure 3 show the Resource utilization rate of MM and PRA. Table 4 shows the cost of MM and PRA. Cost is calculated using formula 5 and 6 for priority tasks. The proposed algorithm minimizes the makespan and cost and maximizes the resources utilization rate.

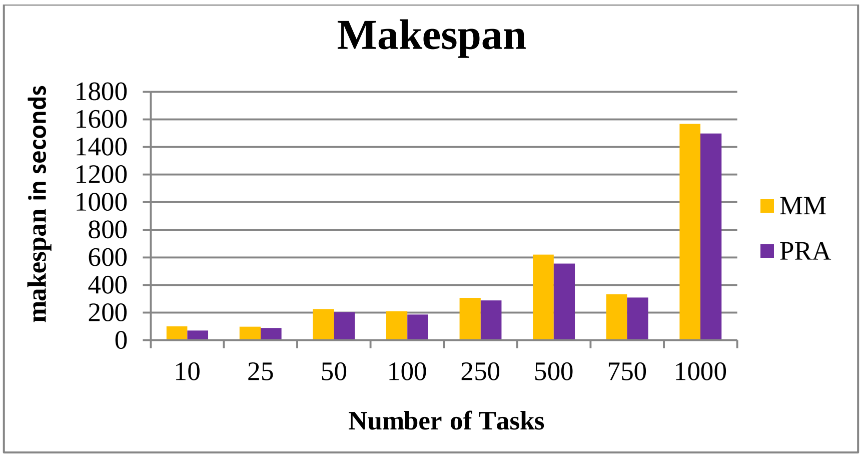

Fig. 2. Makespan of MM and PRA

Table 3. Resource Utilization of different meta-task set in Cloud Sim

No of tasks			No. of resources		Resource Utilization Rate(in %)	
TT	**PT**	**NPT**	**PT**	**NPT**	**MM**	**PRA**
10	7	3	3	2	61	93
25	6	19	3	4	75.5	96
50	18	32	4	6	70.5	95.5
100	41	59	6	8	72.0	96.5
250	59	191	8	14	85.5	98.0
500	461	39	21	6	89.0	98.3
750	729	21	27	4	72.5	92.0
1000	691	309	26	17	94.0	98.5

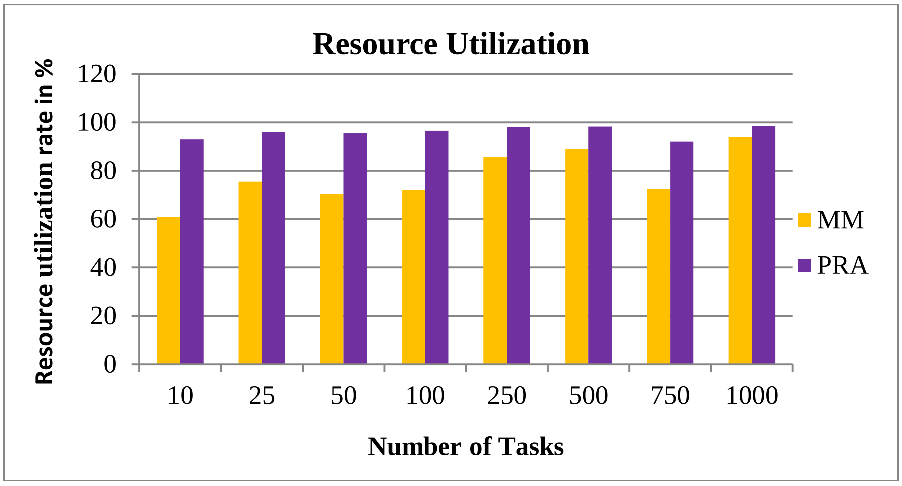

Figure 3. Resource Utilization of MM and PRA

Table 4. Cost of different meta-task set in Cloud Sim

No of tasks			**No. of resources**		**Cost (in rupees)**	
TT	**PT**	**NPT**	**PT**	**NPT**	**MM**	**PRA**
10	7	3	3	2	5.09	5.07
25	6	19	3	4	11.89	11.87
50	18	32	4	6	26.55	26.42
100	41	59	6	8	50.42	50.35
250	59	191	8	14	119.55	119.38
500	461	39	21	6	224.56	224.10
750	729	21	27	4	296.28	295.91
1000	691	309	26	17	415.24	415.00

Table 5. Cost different of PRA over MM

Number of tasks	**Number of Priority tasks**	**Time saved in seconds by PRA over MM**	**Cost saved in rs/hr by PRA ov**
10	7	29.65	2.43
25	6	9.9	7.27
50	18	21.77	21.50
100	41	21.8	11.56
250	59	18.98	32.24
500	461	64.71	25.59
750	729	22.24	59.89
1000	691	70.74	12.21

5. RESULTS

To examine the efficiency of PRA, the experiments have been conducted in cloudsim with the meta-task sets having task and machine heterogeneity. The following parameters are used to compare the results of PRA with Min-Min.

1. ***Makespan***: Makespan is the evaluation of throughput of Cloud. It can be calculated by Eq. (2).

$$CT_{ij}=ET_{ij}+r_j \qquad (2)$$

Where ET_{ij} is time to execute task T_i in resource R_j and r_j is waiting time of T_i

2. Resource Utilization: Utilization rate of resources [11] is evaluated by using Eq. (3).

$$ru = \frac{\sum_{j=1}^{m} ru_j}{m} \qquad (3)$$

ru_j is the average resource utilization rate of resource r_j. It is computed by Eq. (4).

$$ru_j = \frac{\sum(te_i - ts_i)}{T} \qquad (4)$$

Here, te_i and ts_i are the end time and start time of executing t_i on resource m_j respectively and T is the total application time so far.

3. Cost for priority task: Cost of resource is based on MIPS and Memory [17], and is calculated using Eq. (5).

$$TotalCost = \sum_{j=1}^{m} Cost_j \quad (5)$$

Where $Cost_j$ is total execution cost of Resource r_j and is shown in Eq. (6)

$$Cost_j = \sum [ET_j(C+C*0.5))] - [WT_i(C*0.1)]; \quad (6)$$

Were ET_i is execution time of Task T_i, C is cost of fastest resource per second, WT_i is the waiting time of task T_i

Cost of non-priority task is calculated as usual. Overall execution cost is shown in table-4. The time and cost saved by PRA over MinMin is shown in table 5.

The above three metrics are used to conclude that the proposed Priority based Resource Allocation algorithm outperforms the Min-Min algorithm.

6. CONCLUSION

Min-Min algorithm is pertinent if number of small tasks is larger in the non-priority meta-task set. Considering user priority for meta-task set Min-Min algorithm degrades its performance for makespan, cost and resource utilization. To enhance the performance of existing algorithm for tasks with priority, the PRA is proposed. Designing of PRA has good features of Min-Min algorithm and eliminate the limitations. The simulation results from cloudsim for different meta-task sets, consisting of tasks varies from 10 to 1000 tasks, confirm that it has better performance over Min-Min algorithm. It minimizes the makespan and cost and increases the utilization of resources when comparing to Min-Min algorithm. Thus PRA outperforms over Min-Min algorithm. The research may be extended by taking into account of real time applications in cloud.

REFERENCES

[1] Salim Bitam, "Bees life algorithms for job scheduling in cloud computing", International Conference on Computing and Information Technology, 2012.

[2] Anu Gupta, "Cloud Computing Growing Interest and Related Concerns", IEEE 2nd International Conference on Computer Technology and Development (ICCTD) 2010.

[3] Kamalam, G.K., and Dr.V.Murali Bhaskaran," Novel Adaptive Job Scheduling algorithm on Heterogeneous Grid Resources", American Journal of applied Sciences, pp. 1294-1299, 2012.

[4] Harpreet Kaur,Maninder Singh" Review of Various Scheduling Techniques in Cloud Computing" International Journal of Networking & Parallel Computing, Vol.1, Issue 2, November, 2012.

[5] Xiaofang Li, Yingchi Mao, Xianjian Xiao, Yanbin Zhuang," An Improved Max-Min Task-Scheduling Algorithmfor Elastic Cloud" IEEE International Symposium on Computer, Consumer and Control, 2014.

[6] Huankai Chen, Professor Frank Wang, Dr. Na Helian, Gbola Akanmu, "User-Priority Guided Min-Min Scheduling Algorithm For Load Balancing in Cloud Computing", Publication on Research Gate, 2014.

[7] Bhawna Taneja, "An Empirical Study of Most Fit, Max-Min and Priority Task Scheduling Algorithms in Cloud", Computing International Conference on Computing, Communication and Automation (ICCCA2015), IEEE, 2015.

[8] Naseem A.AL-Sammarraie, Mohammed F. Al-Rahmawy, Magdi Z. Rashad, "A Scheduling Algorithm to Enhance the Performance and the Cost of Cloud Services", Computer Engineering and Intelligent Systems, Vol.6, issue 8, 2015.

[9] Er. Rajeev Mangla, Er. Harpreet Singh, "Recovery and User Priority Based Load Balancing in Cloud Computing", International Journal Of Engineering Sciences & Research Technology, ISSN: 2277-9655, Feb-2015.

[10] Bhavisha Kanani, Bhumi Maniyar, "Review on Max-Min Task scheduling Algorithm for Cloud Computing", Journal of Emerging Technologies and Innovative Research (JETIR), Volume 2, Issue 3, March 2015.

[11] Pankajdeep Kaur, Parampreet Singh, "Priority based Scheduling Algorithm with Fast Task Completion Rate in Cloud", Advances in Computer Science and Information Technology (ACSIT), Volume 2, No.10, pp. 17-20, April-June, 2015.

[12] Mokhtar A. Alworafi, Ftyaf Dhari, Asma A. Al-Hashmi, Suresha, A.Basit Darem, "Cost-Aware Task Scheduling in Cloud Computing Environment", Inter Journal of Computer Network and Information Security, Volume 5, pp. 52-59, May-2017.

[13] Hitendre Pal, Bhanvi Rohilla, "A Deadline-Aware Modified Genetic Algorithm for Scheduling Jobs with Burst Time and Priorities", Nature Inspired Computing, Springer, ISBN: 978-981-10-6746-4, pp.55-67, October 2017.

[14] Naoufal Er-raji, Faouzia Benabbou, "Priority Task Scheduling Strategy for Heterogeneous Multi-Datacenters in Cloud Computing",

International Journal of Advanced Computer Science and Application, Volume 8, No. 2, pp. 272-277, 2017.

[15] D.I. George Amalarethinam, S. Kavitha, “Enhanced Min-Min Algorithm for Meta-task Scheduling in cloud Computing”, IJCTA 9(27), pp.85-97, 2016.

[16] Rodrigo N. Calheiros, Rajiv Rangan, Anton Beloglazev, Cesar A.F. De Rose and Rajkumar Buyya, “CloudSim: a tool kit for modeling and simulation of cloud computing environments and evaluation of resource provision algorithms”, SOFTWARE – PRACTICE AND EXPERIENCE, Published online 24 August 2010 in Wiley Online Library.

[17] https://aws.amazon.com/ec2/pricing/on-demand