International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 3 | ISSN : 2456-3307

Multi Attribute Completeness Measure Based Shortest Job First Algorithm in Cloud Environment

H. Krishnaveni*[1], Dr. V. Sinthu Janita Prakash[2]

[1]Department of Computer Science, Cauvery College for Women, Tiruchirappalli, Tamil Nadu, India

[2] Department of Computer Science, Cauvery College for Women, Tiruchirappalli, Tamil Nadu, India

ABSTRACT

The problem of scheduling in cloud environment has been studied in different situations. There are many techniques n presented earlier, but suffers to achieve the scheduling performance. To overcome these problems, an efficient multi attribute completeness measure based task-scheduling algorithm has been discussed in this paper. The method maintains a history of tasks being processed and using that, the method identifies the list of tasks being completed by any resource. For each resource, the method computes the task completeness measure, which is computed using resources available, previous rate of completion and waiting time. Based on computed completeness measure, the method allocates the task to a specific resource using shortest job first. This algorithm reduces waiting time, turnaround time and improves the performance of scheduling.

Keywords : Scheduling, Completeness Measure, Average waiting time and Average turnaround time

I. INTRODUCTION

Cloud computing is utility based computing. It allows the users to access the pool of configurable resources on pay-per-use basis. According to NIST, Cloud Computing is a model for enabling ubiquitous, Convenient, On-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction[1]. It hides the heterogeneity of resources, which are shared dynamically over the cloud and also provides transparency to the users. The main features of Cloud Computing are elasticity, virtualization of resources, self-service provisioning, utility pricing, managed operations, and third party owner-ship.

Cloud Computing follows Service oriented architecture. SaaS, PaaS and IaaS are the main architectural layers of Cloud Computing. Software as a Service layer provides the customers with applications or softwares they demand on payment basis. Cloud Platform as a Service (PaaS), allows the customers to develop, test and deploy their applications on the cloud. The customers can access the resources like processing power, storage capacity, network bandwidth etc. through Infrastructure- as a Service (IaaS).

Cloud computing deployment models are public cloud, private cloud, hybrid cloud and community cloud. Public cloud offers services to the general public through internet. Private cloud is developed for a single organisation which provides services to its users without any charge. Hybrid Clouds are any combination of public, private and community [2].

The organizations which has high amount of data has the problem of storage and it requires much amount to be invested. The small scale organizations could not spend that much amount. So they look for some providers who allocates space on cost. Similarly to execute large scale batch work, it needs high end processors like super computers which cannot be afford by small scale organizations. To support the organizations in performing their tasks, the cloud provider deploys various resources into the cloud and allows the external users to access the resources. The cloud user can submit their task and take results on cost.

The ability to process large scale tasks introduces traffic in submitting the tasks. Many organizations submit their task to the cloud in the same time. The number of tasks submitted will be higher than the number of resources. But in order to process all the requests, they must be ordered over the available resources. Allocating the tasks over the resource is named as task scheduling [3]. There are number of task scheduling algorithms, for example FIFO algorithm schedules the task according to the submission sequence of tasks. Similarly the scheduling can be performed based on their burst time and more. But the efficiency of scheduling algorithm is highly depending on the overall waiting time and makespan.

The rest of the paper is organized as follows: Section 2 reviews the related works. Section 3 presents the proposed algorithm. Section 4 shows the experiment and result. Section 5 provides the concluding remarks.

II. RELATED WORKS

There are a number of algorithms that has been discussed for the problem of task scheduling in cloud environment. This section discuss about some of the methods related to task scheduling in cloud environment.

Amar Ranjan Dash et al [4]. proposed a scheduling algorithm that calculates the priority of processes by using seven priority features and uses dynamic time quantum for scheduling. The results show that the average waiting time and average turnaround time has been reduced when compared to the existing algorithms.

Atul Vikas Lakra et al [5]. developed a Multi-objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization. The main objective is to improve the throughput of the data center and to reduce the execution time and cost without violating the SLA for an application in cloud SaaS environment. It provides the minimum overall execution time and increased throughput than existing algorithms.

Sumit Arora et al[6]. proposed an algorithm, which schedules the tasks based on the standard deviation and mean value to minimize the processing time.

V.Sureshkumar et al [7]. developed a trust based resource selection scheme using BAT-HS for scheduling which results in decreased time to complete tasks by 10.4148% compared to random resource selection having BAT scheduling.

Enhanced Load Balanced Minmin algorithm for Meta Task scheduling proposed by Gaurang Patel et al [8]. uses minmin strategy and reschedules the tasks to reduce makespan and utilize the resources effectively.

Mubarak Haladu et al [9]. proposed an algorithm for resource allocation which reduces overall finishing time and utilizes the resources efficiently. First, the proposed algorithm calculates the expected completion time for all tasks on each resource. It then selects the task with the minimum execution time and the task with maximum execution time on the resource expected to complete in earliest time. Next it computes the difference between the maximum execution time and the minimum execution. If the difference is less or equal to the minimum execution time; it assigns the minimum to the resource that produce it, otherwise assigns the maximum execution time.

Dynamic Fair Priority Optimization Task Scheduling Algorithm proposed by Deepika Saxena et al [10]. has three steps, Constraint- based grouping of tasks, Waited fair priority queue and greedy resource allocation. This method brings fairness at priority level and increases utilization of resources at system level. As a result, the overall execution time and cost has been reduced.

Credit based scheduling in cloud computing environment [11] performs scheduling based on user priority and task length. The high priority task will be scheduled and given more importance.

Ellendula Madhukar et al [12]. proposed an Efficient Scheduling Algorithm for Cloud based on the results of the analysis performed on the log which consists of previous scheduling information for improving the performance of the cloud. The log format referred in this algorithm includes user identity, job identity, size of the job, VM number, VM RAM size and operating system, start time of job, runtime of job, queue time of job.

Performance and Cost Algorithm (PCA) proposed by Naseem A.AL-Sammarraie et al [13]. is a hybrid algorithm that aims to minimize the services cost paid by the user and maximize the profit gained by the provider of services renting. Also it aims to optimize the performance of these services by minimizing the services completion time and maximizing the resource utilization of the resources. It helps the provider to provide the best and most efficient services with highly competitive prices.

III. PROPOSED WORK

For scheduling the tasks over the resource pool, the scheduler considers many factors. The efficiency of scheduling is based on the number of tasks being processed in particular time window, the number of tasks completed successfully, the overall time taken for processing, the average waiting time of all the tasks and so on. By considering many factors the problem of task scheduling can be approached efficiently.

Assumptions:

- All tasks arrive at the same time
- No preemption of tasks
- The resources are heterogeneous in nature

The log format given by Ellendula Madhukar et al [12] acts as the basis for cloud trace concept used in this approach. The proposed method maintains the scheduling trace which includes fields such as taskid, resourceid, turnaround time of tasks, waiting time of tasks and status of completion. Then it splits the trace according to the resources available. For each resource, the method identifies the log and computes the completeness measure. Based on the measure computed, the method schedules the tasks on available resources. The entire phase has been divided into three stages namely preprocessing, multi attribute completeness measure (MACM) calculation and scheduling.

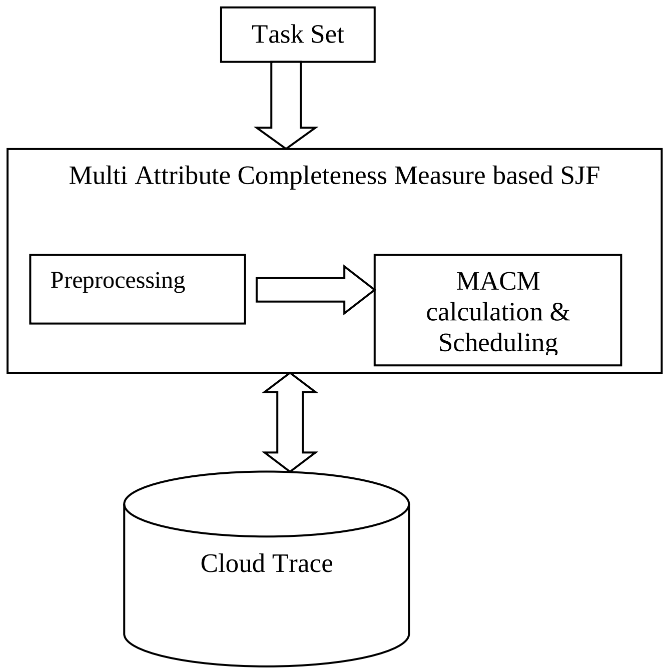

Figure 1. Proposed System Architecture

The Figure.1 shows the architecture of the proposed multi attribute completeness measure based task scheduling algorithm.

A. Preprocessing

This is the first stage of MACM based scheduling. In this stage, the proposed method reads the cloud trace which contains the log about the tasks being executed. Then it identifies the list of resources and splits the trace according to the resource being identified. For each log being identified, the MACMSJFA identifies and verifies the presence of all the properties. If any of the logs has missing values then it will be removed from the log. Preprocessed log will be used to estimate multi attribute completeness measure.

Pseudo code:

1. Read cloud Trace Ct
2. Identify list of resources processed
3. For each Resource Ri
4. Identify list of logs produced by Ri
5. For each log L of Ri
6. If the log is incomplete
7. Remove the log
8. End
9. Add Ri to the Preprocessed Log PL
10. End

The above discussed algorithm identifies and removes the incomplete log from the cloud trace. The inclusion of incomplete traces would affect the scheduling performance. In order to include a trace for the scheduling process, the completeness of the log must be verified. In this stage, the preprocessing algorithm verifies the presence of all the features required in the log considered.

B. Multi Attribute Completeness Estimation

The proposed method, first computes the frequency of resource access. Then the method computes average turnaround time, number of successful completion and average waiting time of the tasks. Using all these measures, it then computes MACM using the formulae given below.

Access frequency (af) = $\frac{size(PL)}{size(CT)} \times 100$ … 3.3.1

Average turnaround time (tat) = $\frac{\sum Turn\ around\ time(taskset)}{No\ of\ tasks}$ ... 3.3.2

Average waiting time (awt) = $\frac{\sum Waiting\ time(taskset)}{No\ of\ tasks}$ ... 3.3.3

Success rate (sr) = $\frac{Number\ of\ success}{Number\ of\ submission}$ ... 3.3.4

MACM = $\frac{af}{sr} \times \frac{tat}{awt}$ ... 3.3.5

To measure the efficiency of any resource considered, the multi attribute completeness measure is estimated. To measure MACM value, the frequency of the resource being used is computed and at each time the makespan time is measured. Similarly the average waiting time of the tasks in choosing the resource is estimated. The success rate is the measure which shows the efficiency of the resource in scheduling. Using all these measures, the MACM value is computed.

C. Scheduling based on MACM

The proposed method performs scheduling by computing the multi attribute completeness measure. First, it preprocesses the log and identifies the list of resources available. For each resource identified, it computes the multi attribute completeness measure. Based on computed measure, it selects the resource and schedules the tasks using shortest job first algorithm.

Pseudo code:

1. Read Cloud Trace
2. Read Task Set TS
3. PL = preprocessing(CT)
4. Identify list of resources available RL

5. For each task Ti
6. For each resource Ri
7. Compute multi attribute completeness measure MACM
8. Choose higher valued resource and apply SJF algorithm
9. Compute turn around time

Based on the computed measure, the proposed method selects the resource to schedule the task. However, the scheduling algorithm chooses the resource which has higher MACM value.

IV. EXPERIMENT AND RESULT

The proposed multi attribute completeness measure based scheduling algorithm has been implemented and evaluated for its efficiency. The proposed method was found to be having a good impact on the problem of scheduling the non-identical resources. The proposed method has produced different time span with different tasks.

The set of tasks referred from [4], given in table.1 are scheduled in two non-identical resources R1 and R2 based on MACM. The MACM value is calculated by considering the previous performance of each resource, which is maintained in the preprocessed log. The proposed algorithm outperforms the existing CSPDABRR, Priority based and Round Robin scheduling algorithms in terms of minimizing the average waiting time and turnaround time. Table I shows the execution time of tasks. Table II displays the comparative results.

Table 1. Tasks Without Arrival Time

Task	Execution Time(sec)
T1	12
T2	32
T3	6
T4	54
T5	83

Table 2. Comparison of CSPDABRR and MACMSJFA

Algorithms	Average Waiting Time (sec)	Average Turnaround Time (sec)
Round Robin	16	53.4
Priority based	16.4	53.8
CSPDABRR	12.4	49.5
MACMSJFA(proposed)	11.2	48.6

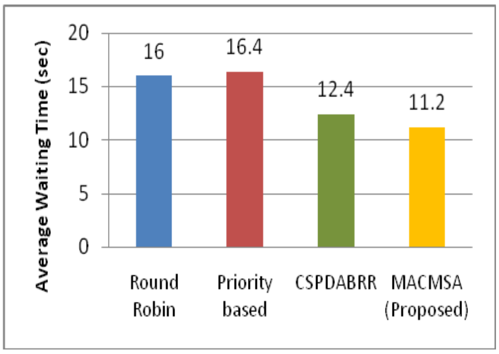

Figure 2. Average Waiting Time Comparison

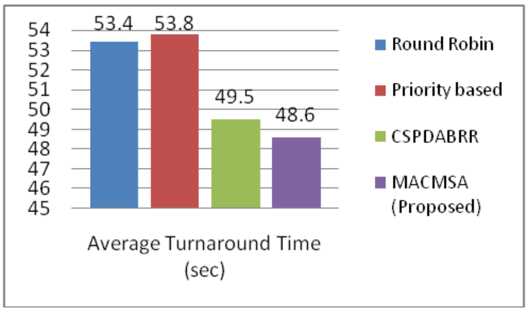

Figure 3. Average Turnaround time comparison

Figure 2 and Figure 3 show that the proposed MACMSJFA algorithm produced better results than the existing algorithms. The result shows that the MACMS algorithm produced average waiting time 11.2 sec and average turnaround time 48.6 sec. The scheduling can also be carried out with 100, 250, 500 tasks and 20, 50, 100 resources.

V. CONCLUSION

In this paper, multi attribute completeness measure based scheduling algorithm has been proposed for the problem of task scheduling. First, it preprocesses the cloud trace to identify the list of resources and splits the log into different classes belong to the resource. Using the log, it computes the multi attribute completeness measure and based on computed measure it schedules the tasks to the resources. The proposed scheduling algorithm reduces the average waiting time and average turnaround time than the existing methods and hence improve the performance of the scheduling. However, in the proposed method, the resources are not utilized efficiently and the load is also not balanced. The performance of scheduling can be improved by adapting divide and conquer approach by utilizing the resources optimally.

VI. REFERENCES

[1]. Avneesh Vashistha, Rabins Porwal, A.K Soni, "A Taxonomy of Scheduling Algorithms for Cloud Computing", International Journal of Computer Science Issues, ISSN (Online): 1694-0784, Vol:12, Iss:1, January 2015.

[2]. Ruby Annette. J , Aisha Banu .W, Shriram, " A Taxonomy and Survey of Scheduling Algorithms in Cloud: based on Task Dependency", International Journal of Computer Applications (0975-8887) Vol: 82 , Iss: 15, November 2013.

[3]. H.Krishnaveni, Dr. V. Sinthu Janita Praksh," A Survey on Task Scheduling Algorithms in Cloud Computing Environment", International Journal of Applied Engineering Research(IJAER), Vol: 10, Iss: 82, Pg: 394 - 400, 2015.

[4]. Amar Ranjan Dash, Sandipta Kumar Sahu, Sanjay Kumar Samantra and Sradhanjali Sabat, "Characteristic Specific Prioritized Dynamic Average Burst Round Robin Scheduling For Uniprocessor And Multiprocessor Environment (CSPDABRR)", International Journal of Computer Science, Engineering and Applications, Vol: 5, Iss: 4, October 2015.

[5]. Atul Vikas Lakra, Dharmendra Kumar Yadav, "Multi-objective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization", Elsevier, ScienceDirect, Procedia Computer Science, Vol: 48, Pg: 107-113, 2015.

[6]. Sumit Arora, Sami Anand, "Improved Task Scheduling Algorithm in Cloud Environment", International Journal of Computer Applications (0975-8887) Vol: 96, Iss: 3, June 2014.

[7]. V. Sureshkumar, M.Aramudhan, "Trust Based Resource Selection in Cloud Computing using Hybrid Algorithm", International Journal of Intelligent Systems and Applications, Vol: 08, Pg: 59-64, 2015.

[8]. Gaurang Patel, Rutvik Mehta, Upendra Bhoi, "Enhanced Load Balanced Minmin algorithm for Meta Task scheduling", Elsevier, ScienceDirect, Procedia Computer Science, Vol: 57, Pg: 545-553, 2015.

[9]. Mubarak Haladu, Joshua Samual," Optimizing Task Scheduling and Resource allocation in Cloud Data Center, using Enhanced Min-Min Algorithm", IOSR Journal of Computer Engineering, Vol: 18, Iss: 4, , Pg 18-25, e-ISSN: 2278-0661, Aug 2015.

[10]. Deepika Saxena, R.K. Chauhan, Ramesh Kait, "Dynamic Fair Priority Optimization Task Scheduling Algorithm in Cloud Computing: Concepts and Implementations", International Journal of Computer Network and Information Security (IJCNIS), Vol:8, Iss:2, Pg:41-48, 2016.

[11]. Antony Thomas, Krishnalal G, Jagathy Raj V P,"Credit Based Scheduling Algorithm in Cloud Computing Environment", Elsevier, Science Direct, Procedia Computer Science, Vol: 46, Pg: 913-920, 2015.

[12]. Ellendula Madhukar, Thirumalaisamy Ragunathan, "Efficient Scheduling Algorithm for Cloud", Elsevier, ScienceDirect, Procedia Computer Science, Vol: 50, Pg: 353-356, 2015.

[13]. Naseem A.AL-Sammarraie, Mohammed .F. Al-Rahmawy, Magdi Z. Rashad, "A Scheduling Enhance the Performance and the Cost of Cloud Services", Computer Engineering and Intelligent Systems, IISTE, Vol:6, Iss: 8, ISSN: 2222-2863 (Online), 2015.