Achieving Service Level Agreement in Cloud Environment using Job Prioritization in Hierarchical Scheduling

Rajkumar Rajavel[1], Mala T[2]

[1] Research Scholar, Department of IST, Anna University, Chennai, Tamil Nadu, India
[2] Assistant Professor, Department of IST, Anna University, Chennai, Tamil Nadu, India
[1] rajkumarprt@gmail.com
[2] malanehru@annauniv.edu

Abstract. One of the challenging issues in Cloud computing Environment is meeting the Service Level Agreement (SLA). The SLA is an agreement signed between the service provider and the service consumer for accessing the service provided by the service provider over the internet. We can investigate the negotiation strategy between the service provider and the service consumer through the third party called a Broker. In many approaches SLA is designed and trusted through the measurement of various non-functional requirements such as response time of job, CPU usage, memory usage and the storage used by the consumer. Main focus of the business process is satisfying the customer need by quick response. In our proposed approach for satisfying the service consumer (customer), parameter such as response time of deadline based job is considered. The response time of the job is affected due to improper scheduling of the job. Therefore a novel hierarchical scheduling with job prioritization is used to give more priority for deadline based jobs. This approach will satisfy the service consumer and meet the SLA by increasing the performance of the scheduling algorithm.

Keywords: Service Level Agreement, service provider, service consumer, negotiation, response time, hierarchical scheduling, job prioritization.

1 Introduction

Cloud computing is a general term for “anything” that can be accessed as the service over the internet. The services can be Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) and Storage as a service [10]. Cloud computing is a new technology which initiate the new way or new trend of computing where the readily available resources are accessed as a service over the internet. The main feature of this cloud computing is “pay-for-use” technique where the service consumer can pay the amount in the online for the number of resource instantly used and the duration of time the resources are accessed. The key properties of the cloud computing is user centric, task centric, powerful, accessible, intelligent and programmable. The concept of cloud computing is the next step to group collaboration. In-order to do this type of project, we need to deploy or house the project to the cloud so that the project can be accessed from anywhere from the internet enabled locations.

The main advantages of the cloud computing are listed as lower cost computer for use, improved performance, lower IT infrastructure cost, lower software cost, instant software updates, fewer maintenance issue, increased computing power, unlimited storage capacity, increased data safety and improved compatibility between operating system. There are certain limitations in cloud environment such as it requires constant internet connection, availability of service, scalable storage, software licensing and etc, which leads to various research activities [6], [7]. In our proposed approach the scenario considered is software licensing which means pay-for-use license. It is an agreement signed between the service provider and the service consumer as a software document with the negotiation between them.

The main purpose of the SLA licensed software document in business model is to ensure the guarantee of both the service consumer and service provider on accessibility of resource and completion of job respectively. Here the service provider has to complete the job (task or request) assigned by the service consumer within the stipulated time as mentioned in the SLA [1], [3]. Similarly the service consumer has to utilize the resource facility with proper CPU speed, Cache memory, number of nodes or Virtual Machine (VM) and the storage as mentioned in the SLA document. Sometimes the service provider may fail to meet the SLA as mentioned in the document because of the unavailability of the resource and overload of the resource due to improper scheduling of jobs. So a novel hierarchical scheduling with multilevel feedback queue is proposed to meet the SLA. The main purpose of the multilevel feedback queue is to preempt the dead line based job for the execution through which we can meet the SLA and can satisfy the consumer. By means of customer satisfaction we can obviously increase the number of cloud user and increase the productivity of the business.

2 Related Work

The important issues in Cloud are performance degradation in the cloud business due to customer dissatisfaction (when the SLA is not meet by the service provider). There are several approaches to meet the SLA between the service provider and the service consumer [5]. In the paper [2], deadline-aware heuristic approach is used which will serve the deadline based job first. It is very difficult to prioritize the deadline based job using a single queue implementation. So in our approach multilevel feedback queue is used for giving more priority for deadline based jobs. The First Come First Served, Shortest Job First and Heuristic Cost Based Scheduling algorithms were implemented and their performances are evaluated [8]. Here the quality of service is measured using the response time of individual jobs. Even if a single job fails to respond within the stipulated time it leads to violation of SLA. Sometime this approach may lead to violation of SLA because it estimate the heuristic cost by using the waiting time and it give more priority to the job which is having less heuristic cost. Main reason here is some jobs might have less heuristic cost and it is not the deadline based job, but this job is given more priority according to this approach. But some deadline based jobs might have more heuristic cost which will suffer by starvation due to less priority over the heuristic cost estimation.

3 Need for Job Prioritization in Hierarchical Scheduling

In the cloud environment to improve the performance, scheduling is done in hierarchical manner by using scheduler in both Cloud Controller (CLC) and Cluster Controller (CC) level. The jobs dispatched by the CLC will be queued in the cluster node. Here the job will be executed one after another in the First Come First Serve (FCFS) basis in the Round Robin (RR) fashion and this may lead deadline based jobs waiting for long time in the queue based on its position in the queue [4]. This situation will obviously leads to SLA violation and thus it dissatisfies the consumer (not meeting the SLA). In the proposed approach hierarchical scheduling is implemented by using job prioritization in both CLC and CC level to avoid SLA violation.

4 Hierarchical Scheduling in the Cloud Environment

An overview of the Cloud Environment Model is shown in Fig1. It consists of service provider, service consumer, cloud resources, SLA document as a negotiation process and a third party for service guarantee. The service provider might be of different people like Amazon EC2, Microsoft Azure, Google App Engine, Google Apps, Salesforce.com and Microsoft Online Services. Some of the services provides by these service providers are IaaS, PaaS, SaaS and Storage as a Service. The service consumers are the end user who can instantly get the huge computational resource by subscribing to the service provider by connecting to the internet. Consumer can instantly get the resource as a service and start running his application. If the application requires more computational power, the resources can be increased on demand by creating the Virtual Machine and also decreases the resource to certain number by destroying the Virtual Machine. Since the cloud resources are elastic in nature either the resource can be increased or decreased based on the needs of the user application running in the cloud environment. SLA document is a contract which specifies a set of application-driven requirement such as contract duration, estimated number of jobs, estimated memory requirement number of resource (VMs).

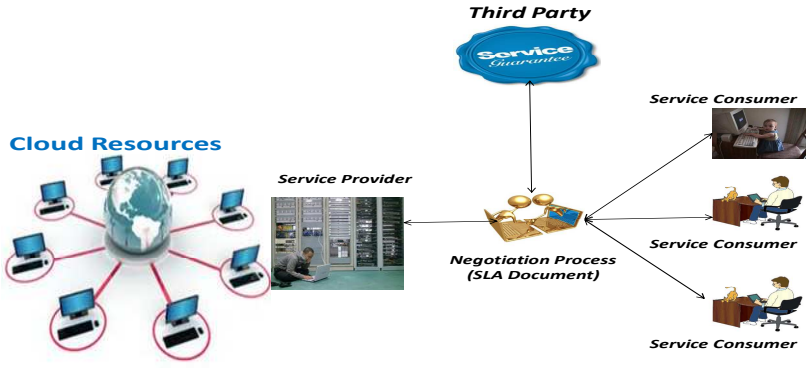

Fig. 1. Overview of Cloud Environment

In this paper Hierarchical Scheduling is proposed in the Cloud Environment as shown in Fig.2 with meta-level scheduling in Cloud Controller (CLC) and local-level scheduling in the Cluster Controller (CC). Service Level Agreement is an agreement signed between the service provider and the service consumer through the negotiation process. If any one of the job is failed to complete within the stipulated time it leads to violation of SLA. We always cannot guarantee the services provided by the service provider and for negotiation with provider we have to depend on the third party. The cloud resources are the group of servers, desktops and PCs which are geographically distributed across the world and connected by the communication media such as wired or wireless. The structure of the cloud resources and its various components such as Cloud Controller, Cluster Controller and Node Controller are organized in the hierarchical structure as shown in Fig. 2.

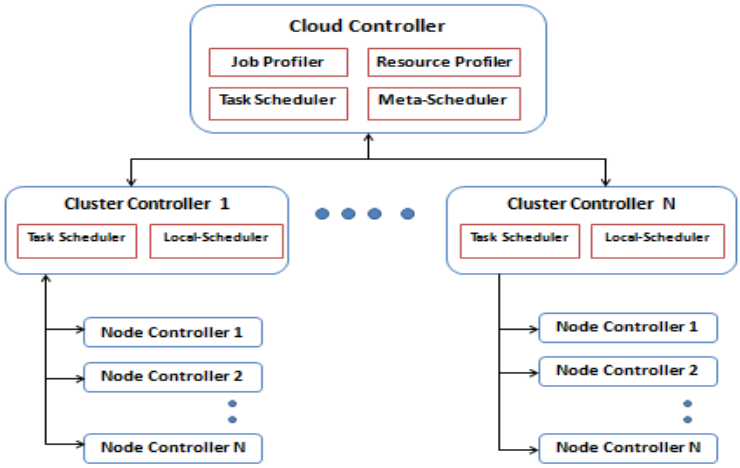

Fig. 2. Structure of Cloud Resources

The main work of the Cloud Controller is to manage and control all the underlying Cluster Controller in the cloud. In addition to that it will schedule all the jobs obtained from the service consumer using the Scheduler component and maintains the result of each job and also update the job status in the Job Profiler component. This Job Profiler will maintain the result and status of all the active jobs which is in execution. It also contains useful information about the non active jobs which are waiting in the queue. The operation of the Cloud Controller will work as shown in Fig. 3. The users request or job will fall into the Cloud Controller Job Queue and for every time interval or instance the Job Puller will pull the job from the Job Queue and sent to the Task Scheduler. Based on the type of job the Task Scheduler will prioritize the job for scheduling process. If the job is deadline or interactive based then it is pushed to high priority queue Q1. In case the job is shortest job it will be pushed to next priority queue Q2, otherwise it will be directly pushed to low priority queue Q3. Since the queue Q1, Q2 and Q3 is connected with the feedback, suppose if one job is moved from the Q1 to meta-scheduler then automatically job from Q2 will be moved to backend of Q1 and similarly the job from Q3 is moved to Q2.

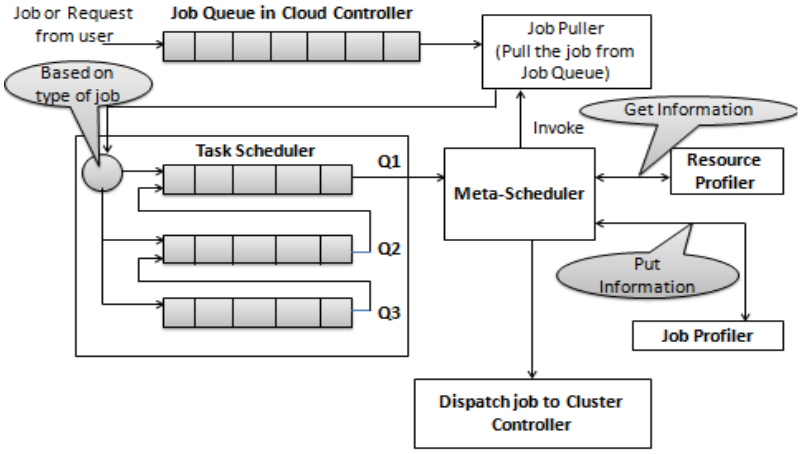

Fig. 3. Operation of Cloud Controller

The Meta-Scheduler will obtain the jobs from Q1 and schedule the job to Cluster Controller by using Load Aware Scheduling Algorithm as shown in Algorithm 1. It will get the resource information from the Resource Profiler and update job information in the Job Profiler during the scheduling process. Suppose if the meta-scheduler find the empty queue in Task scheduler, then it will invoke the Job Puller to pull next set of job from the Job Queue. Finally the job dispatched from the Cloud Controller will fall into the Cluster Controller Job Queue. The Load Aware Scheduling Algorithm in the meta-scheduler will estimate the Load Cost of resource LC(R) by using the equation (1) and (2) as follows,

$$LC(R_i) = QL(R_i) / [\sum NC(R_i)] \quad (1)$$

Where $QL(R_i)$ denotes Queue Load in the resource R_i and it is estimated by using the number of Virtual Machines required by each job as follows,

$$QL (R_i) = \sum [NVM(J_1), NVM(J_2), \ldots , NVM(J_n)] \quad (2)$$

Where $NVM(J_1)$ represents the number of VM required by Job1 and 'n' denotes the number of jobs waiting in the resource R_i.

Algorithm 1. Load Aware Scheduling Algorithm

```
Begin
Get job from Q1
for ( each job)
        Identify the job requirement
        Query the resource profiler for updated resource information
        Estimate the Load Cost for all the resource or cluster controller
        Identify the resource having less Load Cost
        Select the matched resource
        Dispatch the job to matched resource
End
```

The operation of the Cluster Controller will work as shown in Fig. 4. The job is pulled from the Job Queue and based on the type of job the Task Scheduler will prioritize the job to Local-Scheduler. Then the Local-Scheduler will follow the FCFS scheduling and finally dispatch the job to corresponding matched Virtual Machine (VM) present in the Node Controller.

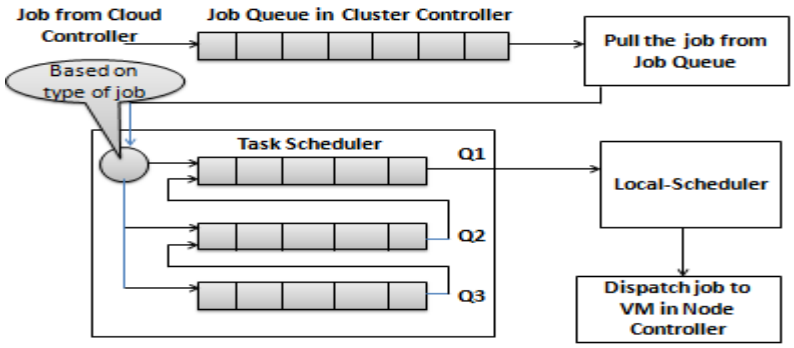

Fig. 4. Operation of Cluster Controller

Since the job is prioritized in hierarchical scheduling, the deadline based jobs will be completed within the stipulated time which leads to meet the SLA made between the provider and the consumer. Job execution and job management in the individual node will be taken care by the Node Controller component. Finally the corresponding output of the job will be dispatched to the end user through the cloud controller component. Based on the response time of the job the third party broker will announce the service guarantee to the job by verifying resource, memory and the estimation cost of the resource used by the user as specified in the SLA document during the negotiation process. If all the jobs are completed within the stipulated time as specified in the SLA document then the result is intimated and the service is guaranteed by the service provider. In case if the job is not completed within the time then the result is specified as SLA violation of service provider. Suppose if the service or resource used by the consumer is more than the specification in the SLA document then it leads to the service consumer or user SLA violation.

5 Experimental Results and Performance Evaluation

In the result phase we have simulated the result using cloudsim toolkit by exploiting five resources and fifty jobs for exactly showing the scheduling algorithm with its response time. Job information present in the job profiler is listed as shown in the Table1. From the table it is clear that only five jobs are deadline based jobs and remaining forty five jobs are non deadline based jobs. All the fifty jobs are submitted to the five available resources only. If you use the FCFS and SJF algorithm all the job requirement cannot be meet, since the deadline based jobs are not given any priority and hence it result in the violation of SLA signed between the service provider and the service consumer [4]. If the same number of job is executed using Load Aware

Scheduling Algorithm and Job prioritization in the hierarchical scheduling will result in the situation where deadline based jobs will be given more priority over the other jobs and it complete the jobs within the stipulated time as specified in the user job requirement. So by using this scheduling algorithm SLA can be achieved easily and satisfy the customer in the cloud business.

Table 1. Job Information

Job ID	Job Size	Completion Time (Deadline of Job)	Execution Time
1 to 10	30 MB	NIL	30 min
11 to 25	20 MB	NIL	30 min
26	30 MB	120 min	30 min
27	20 MB	120 min	40 min
28	30 MB	120 min	50 min
29	20 MB	120 min	40 min
30	30 MB	120 min	40 min
31 to 50	10 MB	NIL	30 min

The results of FCFS, SJF and Load Aware Scheduling (LAS) Algorithm of Hierarchical Scheduling is compared with respect to Completion time of the jobs and the performance measure is also represented as graph as shown in Fig5. In order to show the exact result and its problem definition the results of the deadline based jobs are considered for performance evaluation. From the figure it is clear that only job with ID 26, 27, 28, 29 and 30 are the deadline based jobs.

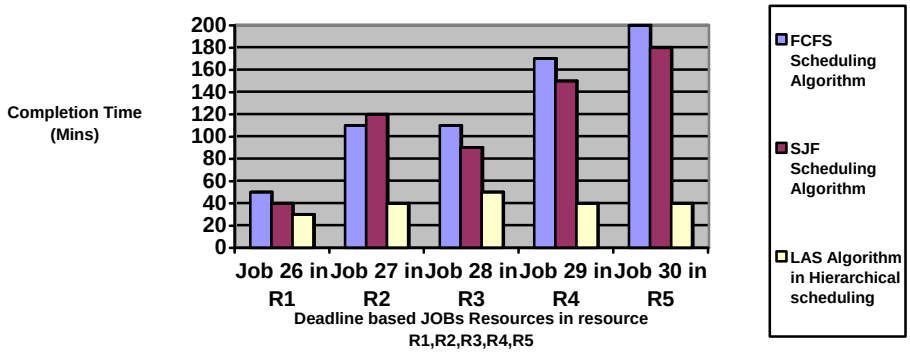

Fig. 5. Performance Evaluation of FCFS, SJF and Multilevel Feedback queue Scheduling Algorithm

The performance of the LAS Algorithm which is proposed in Hierarchical Scheduling works much better than the existing FCFS and SJF Algorithm by completing all the jobs within the stipulated time [9]. Hence from the above performance evaluation it is clear that the proposed Hierarchical Scheduling will increases the performance in the cloud environment and increases the customer satisfaction by quickly responding to the user jobs. By completing all the jobs within the stipulated time SLA is achieved in the cloud.

6 Conclusion and Future Work

In this paper hierarchical scheduling is presented which helps in achieving Service Level Agreement with quick response from the service provider. In our proposed approach Quality of Service metric such as response time is achieved by executing the high priority jobs (deadline based jobs) first by estimating job completion time. Here the priority jobs are spawned from the remaining job with the help of Task Scheduler which increase the performance of the cloud business by quickly responding to the customer. Hence this novel approach provides quick response time comparing to the existing approaches my meeting the SLA in the cloud Environment. In future, multifunctional request handler will be integrated in the cloud controller for handling user jobs in different ways. Here the user can submit the job either through the User Interface provided by the provider or by using the Job Submission Description Language. Thus the multifunctional request handler will helps in handling different way of job submission through SOAP request.

References

1. Artur Andrzejak, Derrick Kondo and Sangho Yi, "Decision Model for Cloud Computing", 18th Annual IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2010.
2. Amril Nazir, Hao Liu and Soren-Aksel Sorensen, "Service Level Agreements in Rental-based Systems", 10th IEEE International Conference on Computer and Information Technology, 2010.
3. Gemma Reig, Javier Alonso and Jordi Guitart, "Prediction of Job Resource Requirements for Deadline Schedulers to Manage High-Level SLAs on the Cloud", 9th IEEE International Symposium on Network Computing and Applications, 2010.
4. Hyun Jin Moon, Yun Chi and Hakan Hacigumus, "SLA-Aware Profit Optimization in Cloud Services via Resource Scheduling", IEEE 6th World Congress on Services, 2010.
5. Hien Nguyen Van, Frederic Dang Tran and Jean-Marc Menaud, "SLA-aware Virtual Resource Management for Cloud Infrastructures", IEEE 9th International Conference on Computer and Information Technology, 2010.
6. J. Oriol Fito, Inigo Goiri and Jordi Guitart, "SLA-driven Elastic Cloud Hosting Provider", 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, 2010.
7. K Hima Prasad, Tanveer A Faruquie, L Venkata Subraminium and Mukesh Mohania, "Resource Allocation and SLA Determination for Large Data Processing Services Over Cloud", IEEE International Conference on Services Computing, 2010.
8. Keerthana Boloor, Rada Chirkova and Yannis Viniotis, "Dynamic request allocation and scheduling for context aware applications subject to a percentile response time SLA in a distributed Cloud", 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010.
9. Keerthana Boloor, Rada Chirkova, Timo Salo and Yannis Viniotis, "Heuristic-based request scheduling subject to percentile response time SLA in a distributed cloud", IEEE Globecom 2010 procedings.
10. Mohammed Alhamad, Tharam Dillon and Elizabeth Chang, "Conceptual SLA Framework for Cloud Computing", 4th IEEE International Conference on Digital Ecosystem and Technologies, 2010.