ISSN 2320-9984

Volume 2 Issue 4 | December 2014 ISSN: 2320-9984 (Online)

International Journal of Modern Engineering & Management Research

Website: www.ijmemr.org

An Efficient Hybrid SJF and Priority based Scheduling of Jobs in Cloud Computing

Devanshu Tiwari
DoCSE, BITS
BITS,
Bhopal (M.P.) [INDIA]
Email : devanshu.tiwari28@gmail.com

Prof. Damodar Tiwari
DoCSE, BITS
BITS,
Bhopal (M.P.) [INDIA]
Email: damodarptiwari21@gmail.com

Abstract**—***Cloud Computing is an important medium through which data can be transmitted via internet. The data to be send by the user is stored at data centers through brokers. But during the designing of cloud environment load balancing and scheduling of resources is an important task, since the various users can request for the same type of resource at the same time, hence scheduling is applied over each requests so that each user can access the resources. Although various scheduling techniques are implemented for the scheduling of resources in cloud such as using FCFS or priority based scheduling, but efficient scheduling is needed since these techniques provides more waiting time and responses time for the users request which needs to be minimize, hence an efficient scheduling technique is implemented for the scheduling of cloud resources.*

Here a hybrid combination of Shortest Job First with Priority based scheduling technique is implemented which reduces the waiting time and turnaround time as well as response time. Scheduling of the cloud resources provides online exchange of information over internet such that the users can use the resources, but scheduling of these resources is important since the chances of deadlock or congestion or latency increases which minimize the performance of the methodology; hence the above issues can be solved using the combination of two scheduling techniques i.e. shortest job first and priority based scheduling. The proposed methodology when implemented improves the performance of the cloud resources. It reduces the latency and communication overhead as well as minimize the communication overhead of the model.

Keywords*:— Scheduling, jobs, data centers, waiting time, FCFS, Priority.*

1. INTRODUCTION

Cloud computing is migrating from theory to practice [1-5]. As a novel large-scale distributed computing paradigm based on virtualized resource pool, cloud computing derived from the ever-increasing interaction and profound development of distributed computing, parallel computing, grid computing, utility computing, web services, etc. If only with accesses to cloud data centers, cloud computing could enable users all over the planet to on-demand leverage a variety of IT-related services based on a pay-per-use model, e.g. infrastructure, platform, software, etc. In addition, cloud computing also corresponds with the basic idea of green computing[6-7].Through elastic and high-scalable management of the resource pool, as well as relaxation of the terminal, cloud computing can conserve much energy and cut the cost of both user's and datacenter's down. It is one of the most promising computing paradigms in the coming low carbon economy.

An application operated on behalf of user in the Inter-Cloud consisting of one or more service requests is sent to the service provider specifying two main constraints, time and cost. Though the response time of a service request cannot be assumed to be accurately estimated, it is most likely that its actual processing time is longer than its original estimate due primarily to delays occurring on the provider's side. This causes the cloud computing are primarily operated by the principle of paying by time, so the service provider want to reduce the delay and improve the quality of their service. Service request scheduling is one of the most important methods to achieve those.

The main goal of job scheduling is to achieve a high performance computing and the best system throughput. Traditional job scheduling algorithms are not able to provide scheduling in the cloud environments. According to a simple classification [8] job scheduling algorithms in cloud computing can be categorized into two main groups; Batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, Jobs are queued and collected into a set when they arrive in the system. The scheduling algorithm will start after a fixed period of time. The main examples of BMHA based algorithms are; First Come First Served scheduling algorithm (FCFS), Round Robin scheduling algorithm (RR), Min–Min algorithm and Max–Min algorithm. By On-line mode heuristic scheduling algorithm, Jobs are scheduled when they arrive in the system. Since the cloud environment is a heterogeneous system and the speed of each processor varies quickly, the on-line mode heuristic scheduling algorithms are more appropriate for a cloud environment. Most fit task scheduling algorithm (MFTF) is suitable example of On-line mode heuristic scheduling algorithm [8].

2. THE FEATURES OF TASK SCHEDULING IN THE CLOUD COMPUTING ENVIRONMENT

In the cloud computing environment, task scheduling and resource assignment have been unified managed by providers through virtualized technology. They have been used to hide and complete users' tasks transparently. Task scheduling becomes more complex because of the transparent and dynamic flexibility of cloud computing system, and the different needs for recourses of different applications. Task scheduling strategies only focus on equity or efficiency will increase the cost of time, space, and throughput and improve the quality of service of the entire cloud computing at the same time. The characteristics of the task scheduling in the cloud computing environment are as follows:

Task scheduling caters to a unified resources platform.

As cloud computing using the virtualized technology, we abstracting the underlying physical resources (all types of hosts, workstations or even PC, etc.) as a unified resource pool, and shielding heterogeneous, supply the upper use. It mainly distributes in a large number of distributed computers, and supplies the use of resources in the form of a data center.

Task scheduling is global centralized.

As cloud computing is a computing model which supply the centralized resource by the mirror service to multiple distributed applications, and this mirroring deployment can make heterogeneous procedures' executing of interoperate become easier, which used to be difficult to deal with. Therefore, virtualized technology and mirroring services make the task scheduling of cloud computing achieve a global centralized scheduling.

Each node in the cloud is independent.

In cloud computing, the internal scheduling of every cloud node is autonomous, and the schedulers in the cloud will not interfere with the scheduling policy of these nodes.

The scalability of task scheduling.

The scale of resources supply from cloud provider may be limited in early stages. With the addition of a variety of computing resources, the size of the abstract virtual resources may become large, and the application demand continues increasing. In the cloud, task scheduling must meet the scalability features, so that the throughput of the task scheduling in the cloud may not be too low.

Task scheduling can be dynamically self-adaptive.

Expanding and shrinking applications in the cloud may be necessary depend on the requirement. The virtual computing resources in cloud system may also expand or shrink at the same time. The resources are constantly changing, some resources may fails, and new resources may join in the clouds or restart.

The set of task scheduling.

Task scheduling is divided into two parts: one is used as a unified resource pool scheduling, and primarily responsible for the scheduling of applications and cloud API; the other is for the unified port resource scheduling in the cloud, for example, Map Reduce task scheduling. However, each scheduling consists of two two-way processes: scheduler leases resource from cloud, scheduler callbacks the requested resources after use. The former process is scheduling strategy and the latter one is callback strategy [9, 10]. The combination of the scheduling and callback resource strategy is the set of task scheduling [11].

3. THE TARGET OF TASK SCHEDULING IN CLOUD ENVIRONMENT

The task scheduling goals of Cloud computing is provide optimal tasks scheduling for users, and provide the entire cloud system throughput and QoS at the same time. Specific goals are load balance, quality of service (QoS), economic principle, the optimal operation time and system throughput [10, 11].

Load balance

Load balancing and task scheduling has close contacts with each other in the cloud environment, task scheduling mechanism responsible for the optimal matching of tasks and resources [12]. Because of the pertinency of task scheduling algorithm, load balancing become another important measure in the cloud. Since load balancing state level two loads in task scheduling under cloud computing environment: the first stage is the virtual machine load, the second one is the resource layer load [10].

Quality of Service

The cloud is mainly to provide users with computing and cloud storage services, resource demand for users and resources supplied by provider are performing in the form of quality of service. When task scheduling management comes to task allocation, it is necessary to guarantee the resources QoS.

Economic Principles

Cloud computing resources are widely distributed throughout the world. These resources may belong to different organizations. They have their own management policies. As a business model, cloud computing, according to the different requirements, provides relevant services. So the demand charges are reasonable. Market economy drives task scheduling and resource management, we must make sure their benefit both (consumer and provider) so that the cloud computing can move more and more further [9, 11].

The best running time

Primarily for applications, tasks can be divided into different categories according to the needs of users, and then set the best running time on the basis of different goals for each task. It will improve the QoS of task scheduling indirectly in a cloud environment.

The throughput of the system

Mainly for cloud computing systems, throughput is a measure of system task scheduling optimizing performance, and it is also a target which has to be considered in business model development. Increase throughput for users and cloud providers would be benefit for them both. [9-12].

4. RESOURCE MANAGEMENT AND TASK SCHEDULING IN CLOUD COMPUTING ENVIRONMENT

Cloud computing technology provides the method of sharing basic infrastructure. It brings the computing resources and storage resources in different geographical positions into a resource pool through virtual technology. Users need to apply before using it, and we need to release resources after using it so that the resources can be reused. In this way, the cloud computing center can provide high-performance computing resources and huge storage resources which are simple and low cost.

With further development and maturity of cloud computing technology the features of efficient, flexible, and customizable provide a new way to solve the problems encountered in the operation process of the scientific workflow. When using the cloud platform, researchers need to upload resources data sets to the cloud computing platform. Because the scale of resources set may be very large, there are bandwidth limitations between different parts and some set of resources can only be stored in the specified resource center, researchers cannot upload all the set of data resources to the same resource center or upload all the set of data resources to every resource center. While they need to upload different set of data resources to different data resource centers, so that the tasks of scientific workflow can be executed in parallel. Because there are strong data resource dependencies between scientific workflow, their implementation often requires frequent transfer and access to the resource center. Unreasonable resource data placement and task scheduling strategies will lead to the excessive of transmission volume and traffic volume easily. On one hand, it increases the user fees for the use of cloud resources, on the other hand, it seriously impact the implementation efficiency of the scientific workflow. So studying an effective and fair task scheduling algorithm in a cloud environment is not only important in resources transmission and reducing the transmission of user fees, but also important in enhancing the implementation of the performance and user satisfaction.

Queries about cloud computing environment

The cloud computing environment is famous for its business model. A variety of tasks that users demand, a unified solution, task scheduling remains key issues. There are analogies between the allocating resources in cloud computing system and distribution of social income wealth: The resources provided by foundation facilities manufacturers are equivalent to overall social wealth. The demands of different users are said of different task forms, which can abstractly as social individual. The resources volume paid by user can be seen as reward that social individual gets by labor. On the basis of labor differences, they distribute different wealth.

In cloud computing, we know New Berger model of justice distribution and efficiency (time and cost game) which is mainly based on following several points:

1. Cloud computing offers available computing and resources storage for a variety of users and enterprises. Therefore, the cloud computing requires the general resource-allocation policy to meet the needs of different users for the allocation of resources, and achieves higher quality of service. In cloud computing environments, by introducing New Berger fair and justice distribution theory of the actual demand, resource allocation can be assigned as social

distribution with the nature of automatic regulation.

2. Traditional task scheduling algorithm focuses on efficiency or cost. It has pertinence for tasks of Specific style and targets for specific types of tasks such as to target the least finishing time, the most optimum availability, and the least cost. These scheduling policies have better efficiency or better cost advantages, but can cause uneven loading, unilateral advantages of efficiency and cost. The expectations of enterprises integrated QoS are not balanced, that means service requirement quality of task scheduling in cloud environment cannot meet the expectations of users. Therefore, it's more important that the efficiency and cost of task scheduling model are balanced. Task scheduling in cloud computing environment should not only meet the balance between efficiency and cost, but also meet the equitable resources distribution. New Berger model emphasizes these two aspects both [13, 14].

3. Divide user's tasks according to the QoS, users have a clear direction of resources service. The game between efficiency and cost is based on meeting users' benefits or fee requirement and then seeking optimal value or equilibrium point. And finally achieve double-win in user efficiency and cost. We call this process efficiency optimization

4. Cloud computing uses virtualized technology to pack resources and then supply for users. These new traits require us to establish a link between users and virtual resources. And we need to develop new applicable task scheduling and resource mapping mechanism.

5. The level of QoS: To enhance the overall QoS is same as improve customer satisfaction; the main method is using the strategy which is fair and satisfactory. So benefit comes first, and we consider the fairness and costs at the same time [13-15].

5. GUIDELINES OF SCHEDULING

Job Scheduling is used to allocate certain jobs to particular resources in particular time. In cloud computing, job- scheduling problem is a biggest and challenging issue. Hence the job scheduler should be dynamic. Job scheduling in cloud computing is mainly focuses to improve the efficient utilization of resource that is bandwidth, memory and reduction in completion time. An efficient job scheduling strategy must aim to yield less response time so that the execution of submitted jobs takes place within a possible minimum time and there will be an occurrence of in-time where resources are reallocated. Because of this, less rejection of jobs takes place and more number of jobs can be submitted to the cloud by the clients which ultimately show increasing results in accelerating the business performance of the cloud. There are different types of scheduling based on different criteria, such as static vs. Dynamic, centralized vs. Distributed, offline vs. Online etc are defined below:

1. ***Static Scheduling:*** Pre-Schedule jobs, all information are known about available resources and tasks and a task is assigned once to a resource, so it's easier to adapt based on scheduler's perspective [16].
2. ***Dynamic Scheduling:*** Jobs are dynamically available for scheduling over time by the scheduler. It is more flexible than static scheduling, to be able of determining run time in advance. It is more critical to include load balance as a main factor to obtain stable, accurate and efficient scheduler algorithm [16].
3. ***Centralized Scheduling:*** As mentioned in dynamic scheduling, it's a responsibility of centralized / distributed scheduler to make global decision. The main benefits of centralized scheduling are ease of implementation; efficiency and more control and monitoring on

resources. On the other hand; such scheduler lacks scalability, fault tolerance and efficient performance. Because of this disadvantage it's not recommended for large-scale grids [17].

4. ***Distributed / Decentralized Scheduling:*** This type of scheduling is more realistic for real cloud despite of its weak efficiency compared to centralized scheduling. There is no central control entity, so local schedulers' requests to manage and maintain state of jobs' queue [19].
5. **Preemptive Scheduling:** This type of scheduling allows each job to be interrupted during execution and a job can be migrated to another resource leaving its originally allocated resource, available for other jobs. If constraints such as priority are considered, this type of scheduling is more helpful [18].
6. ***Non Preemptive Scheduling:*** It is a scheduling process, in which resources are not being allowed to be re-allocated until the running and scheduled job finished its execution [18].
7. ***Co-operative scheduling:*** In this type of scheduling, system have already many schedulers, each one is responsible for performing certain activity in scheduling process towards common system wide range based on the cooperation of procedures, given rules and current system users [17].
8. ***Immediate / Online Mode:*** In this type of scheduling, scheduler schedules any recently arriving job as soon as it arrives with no waiting for next time interval on available resources at that moment [20].
9. **Batch / Offline Mode:** The scheduler stores arriving jobs as group of problems to be solved over successive time intervals, so that it is better to map a job for suitable resources depending on its characteristics [20].

6. TASK SCHEDULING MODEL

Task scheduling model [24] is based on the concept of value function which is shown in figure1. Value function includes deadline, reward, decay, bottom line and penalty. Deadline represents time bottom line of the task to reflect time demand of user expecting to complete the task before this time and when the service provider fails to meet the needs of the user time, provider should pay compensation. Reward represents the cost that users pay for service of service provider and when provider meets the demands of user, provider gets the reward. Decay represents the ratio of compensation paid by provider. Bottom line represents time corresponding to the highest compensation offered by provider and with the time going on, compensation increases gradually, at this moment, compensation get to the maximum value and does not increase any more. Penalty represents the highest compensation offered by provider and with the time going on; compensation increases gradually, at this moment, compensation get to penalty and does not increase any more.

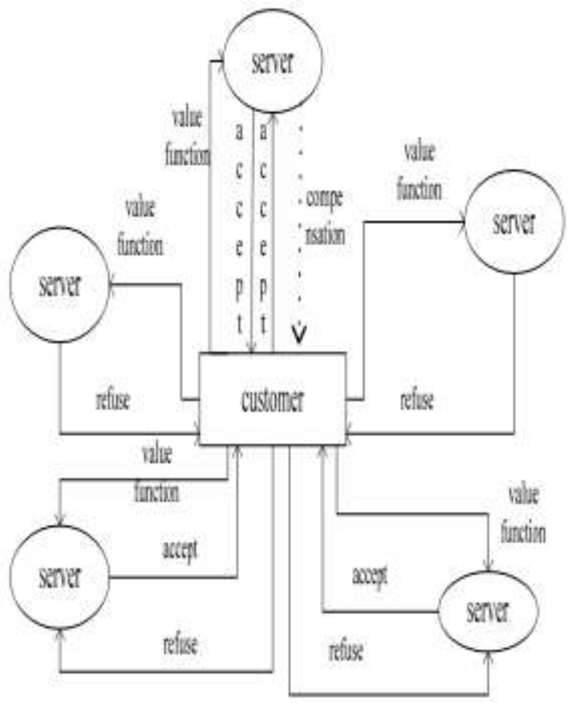

Figure 1. Task Scheduling model.

The above figure shows the basic task scheduling model. Here in this model is given a number of servers which can accept and refuse the values of jobs from the customer. The various jobs to be accessed by the user or customer is queried to the server, where the server checks the validity of the job in the scheduling pool, if the job is allotted accept function calls otherwise refuse.

The task scheduling algorithm consists of some functions so that the task is scheduled by the servers which are request by the user.

Accept Function: The accept function is used here between user request to the server. The various request or task given by the user is send to the server where server checks the scheduling of the task and on the basis accept the requests.

Refuse Function: This function gets call when a user is requested for some task but server refuses due to time or already allotted resources. The server once refuses the request of the user will be requested again by the user.

Value Function: This function is used to send the value to the server by the user once the request is accepted.

7. LITERATIRE SURVEY

Following Priority based Job scheduling techniques are currently prevalent in clouds:

In this paper [30] they have proposed a priority based job scheduling algorithm which can be applied in cloud environments they have named it "PJSC". Also we have provided a discussion about some issues related to the proposed algorithm such as complexity, consistency and finish time. Priority of jobs is an important issue in scheduling because some jobs should be serviced earlier than other those jobs can't stay for a long time in a system. A suitable job scheduling algorithm must consider priority of jobs. It is a multi-criteria decision-making (MCDM) and multi-attribute decision-making (MCDM) model. Basically architecture of AHP is consisted of three levels which are objective level, attributes level and alternatives level respectively. Result of this paper indicates that the proposed algorithm has reasonable level of complexity.

In this paper [31] we have proposed a novel scheduling heuristic by considering QoS factor in scheduling and have proposed some modifications using existing Sufferage heuristic and Min-min heuristic. Additionally, all of these scheduling approaches consider neither both the different levels of user tasks' Qos requests nor the resource properties of dynamic and heterogeneity in cloud computing environment. A QoS guided Sufferage-Min heuristic model, mainly inheriting from the Sufferage algorithm and Min-min algorithm, is presented in this paper after comparing and analyzing different heuristic algorithms. This model is composed of a Sufferage-Min heuristic algorithm and some Qos guided scheduling strategies considering the QoS requirements, low execution complexity, and the dynamic and heterogeneity resource properties in cloud computing environment. We have compared our proposed scheme to other scheme based on heuristic algorithm with a particular example and proved that the proposed scheduling heuristic had a significant performance gain in terms of reduced makespan. QoS is an extensive concept and varies from different research application.

Mayank Mishra et al. [32] in his paper has told that, the users of cloud services pay only for the amount of resources (a pay-as-use model) used by them. This model is quite different from earlier infrastructure models, where enterprises would invest huge amounts of money in building their own computing infrastructure. Typically, traditional data centers are provisioned to meet the peak demand, which results in wastage of resources during non-peak periods. To alleviate the above problem, modern-day data centers are shifting to the cloud. The important characteristics of cloud-based data centers are making resources available on demand. The operation and maintenance of the data center lies with the cloud provider. Thus, the cloud model enables the users to have a computing environment without investing a huge amount of money to build a computing infrastructure. This provides ability to dynamically scale or shrink the provisioned resources as per the dynamic requirements. Fine grained metering. This enables the pay as- use model, that is, users pay only for the services used and hence do not need to be locked into long-term commitments. As a result, a cloud-based solution is an attractive provisioning alternative to exploit the computing- as-service model.

Venkatesa Kumar. V and S. Palaniswami [33], in their paper, have proposed the overall resource utilization and, consequently, reduce the processing cost. Our experimental results clearly show that our proposed preemptive scheduling algorithm is effective in this regard. In this study, we present a novel Turnaround time utility scheduling approach which focuses on both the high priority and the low priority takes that arrive for scheduling.

Vijindra and Sudhir shenai [34] in their paper, have presented an algorithm for a cloud computing environment that could automatically allocate resources based on energy optimization methods. Then, we prove the effectiveness of our algorithm. In the experiments and results analysis, we find that in a practical Cloud Computing Environment, using one whole Cloud node to calculate a single task or job will waste a lot of energy, even when the structure of cloud framework naturally support paralleled process. We need to deploy an automatic process to find the appropriate CPU frequency, main memory's mode or disk's mode or speed. We have also deployed scalable distributed monitoring software for the cloud clusters.

Liang Luo et al.[35] in their paper, have discussed about, a new VM Load Balancing Algorithm is proposed and then implemented in Cloud Computing environment using CloudSim toolkit, in java language. In this algorithm, the VM assigns a varying (different) amount of the available processing power to the individual application services. These VMs of different processing powers, the tasks/ requests (application services) are assigned or allocated to the most powerful VM and then to the lowest and so on. we have optimized the given performance parameters such as response time and data processing time, giving an efficient VM Load Balancing algorithm i.e. Weighted Active Load Balancing Algorithm in the Cloud Computing environment.

Xin Lu, Zilong GU [36], in their paper have discussed that, by monitoring performance parameters of virtual machines in real time, the overloaded is easily detected once these parameters exceeded the threshold. Quickly finding the nearest idle node by the ant colony algorithm from the resources and starting the virtual machine can bears part of the load and meets these performance and resource requirements of the load. This realizes the load adaptive dynamic resource scheduling in the cloud services platform and achieves the goal of load balancing.

Zhongni Zheng, Rui Wang [37] did the research of using GA to deal with scheduling problem in the cloud, we propose PGA to achieve the optimization or sub-optimization for cloud scheduling problems. Mathematically, we consider the scheduling problem as an Unbalanced Assignment Problem. Future work will include a more complete characterization of the constraints for scheduling in a cloud computing environment, improvements for the convergence with more complex problems.

Lu Huang, Hai-shan Chen [38] also presented system architecture for users to make resource requests in a cost-effective manner, and discussed a scheduling scheme that provides good performance and fairness simultaneously in a heterogeneous cluster, by adopting progress share as a share metric. By considering various configurations possible in a heterogeneous environment, we could cut the cost of maintaining such a cluster by 28%. In addition, we proposed a scheduling algorithm that provides good performance and fairness simultaneously in a heterogeneous cluster. By adopting progress share as a share metric, we were able to improve the performance of a job that can utilize GPUs by 30% while ensuring fairness among multiple jobs.

Sunita Bansal, Bhavik Kothari, Chitranjan Hoda [39] proposed a novel grid scheduling heuristic that adaptively and dynamically schedules task without requiring any prior information on the workload of incoming tasks. This models the grid system in the form of a state – transition diagram with job replication to optimally schedule jobs. This algorithm uses prediction information on processor utilization. In this algorithm they uses concept of job replication that is, a job can be replicated to other resource if that resource completes execution of current job than the resource it is currently allocated. This algorithm uses two types of queue namely, Waiting Queue and Execution Queue. This approach is based on exploiting information on processing capability of individual grid resources and applying replication on tasks assigned to the slowest processors. The approach facilitates replication of tasks, and also assigned to execute on slower machines, on machines with higher processing capacity. In this approach the communication costs are ignored. Experimental results show the better performance of this approach compared to traditional round robin algorithm.

Li Yang, ChengSheng Pan, ErHan Zhang, HaiYan Liu [40] proposed one kind of weighted fair scheduling algorithm. It is based on strict rob priority class which adds an absolute priority queue based on the foundation of based class weighted fair scheduling algorithm (CBWFQ). This algorithm covers the disadvantage of traditional weighted fair scheduling algorithm. Weighted Fair Scheduling algorithm differentiates the services of all active queues on the basis of weight of each business flow. When a new job arrives the classifier classifies the jobs into categories. Then buffer is checked for each category and if buffer is not overloaded then job is stored in the buffer otherwise job is dropped. Each job enters a different virtual queue. Weight, Dispatch, Discard and Rob are four main rules of this algorithm. The main advantage of this algorithm is that it has introduced the rob rule together with dropping rule. Experiments are done on NS-2 software to simulate SRPQ-CBWFQ algorithm. This new algorithm combined buffer management and queue scheduling and only guarantees low delay of real time applications. It also gave consideration to fairness and better utilization of buffers. This algorithm has two great advantages of bandwidth allocation and delay without throughput reducibility.

Shamsollah Ghanbari, Mohamed Othman [41] presented a novel approach of job scheduling in cloud computing by using mathematical statistics. This algorithm considers the priority of jobs for scheduling and named as priority based job scheduling algorithm. It is based on multiple criteria decision making model. A pair wise comparison based on multiple criteria and multiple attributes method was first developed by Thomas Saaty [61] in 1980 and named as Analytical Hierarchy Process (AHP). Consistent Comparison Matrix is the foundation of AHP, so to use the concept of AHP comparison matrices are computed according to the attributes and criteria's accessibilities.

In this algorithm, each job requests a resource with determined priority. So comparison matrices of each jobs according to resources accessibilities is computed and also comparison matrix of resources is computed. For each of the comparison matrices priority vectors (vector of weights) are computed and finally a normal matrix of all jobs is computed named as Δ. Likewise, normal matrix of all resources is also computed and name of that matrix is γ. The next step of the algorithm is to compute Priority Vector of S (PVS), where S is set of jobs. PVS is calculated by multiplying matrix Δ with matrix γ. The final step of the algorithm is to choose the job with maximum calculated priority, so a suitable resource is allocated to that job. The list of jobs is updated and the scheduling process continues till all the jobs are scheduled to suitable resource. Experimental results indicate that the algorithm has reasonable complexity. Also there are several issues related to this algorithm such as complexity, consistency and finish time.

Agent based Priority Heuristic for Job Scheduling on Computational Grids [42] this algorithm presents an agent based job scheduling for effective and efficient execution of user jobs. This considers QoS parameters like waiting time, turnaround time, and response time, total completion time, etc. Priorities are assigned to the jobs under different classifications. Agent based Heuristic Scheduling (AHS) uses task agent for job distribution to achieve optimum solution. Task agent receives jobs from users and distributes them among different prioritize global queues based on user levels. AHS uses agent based job distribution strategy at global level for optimal job distribution based on user levels and job priorities at local levels for efficient and effective execution of jobs. For different global queues, priorities are defined as threshold levels for assigning jobs to global queues. If jobs have same priorities then jobs having minimum run time executes first otherwise First Come First Serve (FCFS) algorithm is used. AHS has optimal performance with respect to QoS parameters.

Design, Development and performance analysis of Deadline based Priority Heuristic for Job scheduling in a Grid [43]: A modified prioritized deadline based scheduling algorithm (MPDSA) is proposed using project management algorithm for efficient job execution with deadline constraint of user's jobs. MPDSA executes jobs with closest deadline time delay in cyclic manner using dynamic time quantum. It assumes each job to be described by its process_id, burst_time, arrival_time and deadline. Time quantum is assigned by computing LCM of all burst times. Then the jobs having minimum time delay is selected for execution. If jobs have same time delay then First Come First Serve (FCFS) algorithm is used for scheduling. Jobs are pre-empted based on time quantum and if a job completes its execution before time quantum, that job is deleted from queue. This algorithm satisfies system requirements and supports scalability under heavy workloads.

A two-stage-priority-rule-based algorithm for robust resource-constrained project scheduling [43]: The algorithm solves the resource-constrained project scheduling problem (RCPSP). This algorithm presents a two-stage algorithm for robust resource-constrained project scheduling. First stage solves the RCPSP for minimizing make span by using a priority-rule-based heuristic. Second stage is intended to find most robust schedule with a make span not larger than threshold value found in first stage. Both the stages are referred as two phases. In phase I, each iteration has three steps:

1. Priority values issued from selected priority rule.
2. Random-biased selection of eligible activities according to their selection probabilities.
3. Selected activities are scheduled to the resources.

7. PROBLEM STATEMENT

Priority based Job Scheduling is a technique of providing scheduling to the jobs that are requests to the broker. Priority based job scheduling is a technique which provides scheduling of resources, users requests, virtual memories and data centers. Priority based Job Scheduling provides high complexity and reduces time complexity and provides CPU utilization.

Scheduling of jobs in cloud environment involves improved CPU Utilization Time and less waiting time as well as less turnaround time.

The existing techniques used for the scheduling of jobs in cloud computing such as First Come First Serve suffers from problem since it process the jobs firs which takes more time and other process that has less burst time have to wait till the other finishes. Also Shortest Job First suffers from problem.

Priority based scheduling used for the scheduling of jobs provides less waiting time but this scheduling techniques also suffers from the problem where jobs are scheduled on the basis of priority of jobs in cloud, but can't perform when priority is more but burst time is more which result in a less CPU utilization and takes more waiting time as compared to the normal execution.

Cloud providers often have several powerful servers and resources in order to provide appropriate services for their users but cloud is at risk similar to other Internet-based technology. In Cloud Computing users can access various resources in cloud computing using scheduling techniques but the techniques implemented reduces the waiting time very less.

8. PROPOSED METHODOLOGY

The proposed methodology implemented here for the scheduling of the resources in cloud computing using hybrid combinatorial method of priority based Shortest Job First.

First Come First Server

1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.
3. Schedule allocation of resources 'Ri' to Users 'Ui' using First Come First Server algorithm.
4. The Users that request the Resource First is allocated the resource first.
5. It is also called FIFO.

Shortest Job First

1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.
3. Schedule allocation of resources 'Ri' to Users 'Ui' using Shortest Job First algorithm.
4. Scheduling of Resources can be done on the basis of their shortest duration time.

Priority Based Scheduling

1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.
3. Schedule allocation of resources 'Ri' to Users 'Ui' using priority based scheduling algorithm.
4. Scheduling of Resources can be done on the basis of priority of resources.

Proposed Scheduling

The proposed methodology implemented here consists of combining priority based scheduling and Shortest Job First Scheduling.

1. First of all create Cloud Simulation environment.
2. set the number of user 'Ui' and a Number of resources 'Ri' to access.
3. Schedule allocation of resources 'Ri' to Users 'Ui' using priority but checks whether there Source takes shortest duration time to execute or not.

Algorithm

The proposed methodology uses the combination of two scheduling techniques Shortest Job First and Priority based Scheduling. The processing can be applied for the public as well as Hybrid Cloud. The formal algorithms steps of the proposed methodology are given below:

Table 1 Various Notations used in Algorithm

Notation	Description
Ui	Various Users of the cloud
DCi	Data Centers
UBi	Broker of the cloud
Bi	Burst Time for the Job Ji
Ji	Sequence of Jobs
Ti	Process time
Pi	Priority of Job Ji

1. If 'N' is the number of requests to send from 'Ui; users of the cloud 'C' with 'Bi'

burst time of each of the user 'Ui' to the data Centers 'DCi' through brokers 'UBi'.

2. If 'T1, T2, …………Tn' is the various burst time from various users "Ui' for the request of the Jobs 'Ji'.
3. If 'P1,P2,…………Pn' be the priority of various jobs 'Ji' for the request.
4. Check the 'Ji' having highest priority from 'Pi'.
5. Now also check the burst time 'T' of the job 'Ji' having highest priority 'Pi'.
6. If burst time 'Ti' is very less then execute the scheduling of the job 'Ji'.
7. Otherwise the job having shortest burst time 'Ti' is executed.

The figure 2 shows flow chart of the proposed methodology. It consists of number of inputs from the users of the cloud along with their burst time. Now the jobs are sending to the broker for scheduling. Now at the broker side burst time of each of the user is checked and their priority, if the priority of the user is more than that job is executed first but if the highest priority job contains more burst time than this job can't be executed first.

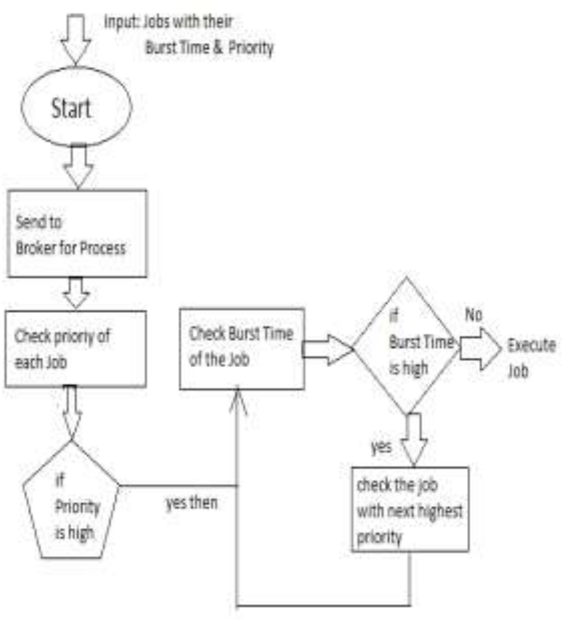

Figure 2. Flow Chart of the proposed methodology

9. SIMULATION AND RESULT ANALYSIS

We are developing an application which is provided two factor authentications. For developing this application, we are using JDK 1.6 that is JAVA developing kit and the simulation is performed with the help of CloudSim (It is a simulator tool which provides a framework for the simulation and modelling of cloud computing services and infra structure. This tool is available in various versions in from which we are using CloudSim2.1 tool).

The Graph shown below is the processing time to access the resources for the users of the cloud. Here comparison between First Come First Serve, Shortest Job First and Proposed techniques for various resources is done.

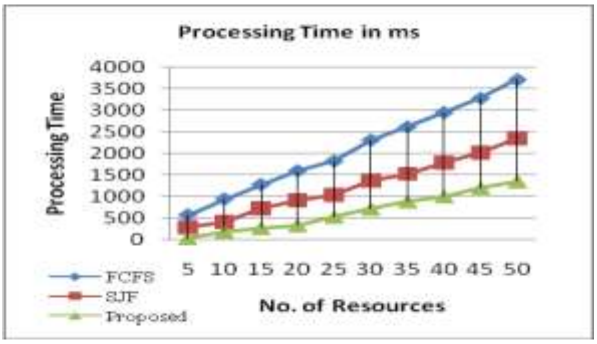

Figure 3 Processing Time in MS

The Graph shown below is the Waiting time to access the resources for the users of the cloud. Here comparison between First Come First Server, Shortest Job First and Proposed techniques for various resources is done.

Figure 4: Waiting Time in MS

The Graph shown below is the Response time to access the resources for the users of the cloud. Here comparison between First Come First Server, Shortest Job First and Proposed techniques for various resources is done.

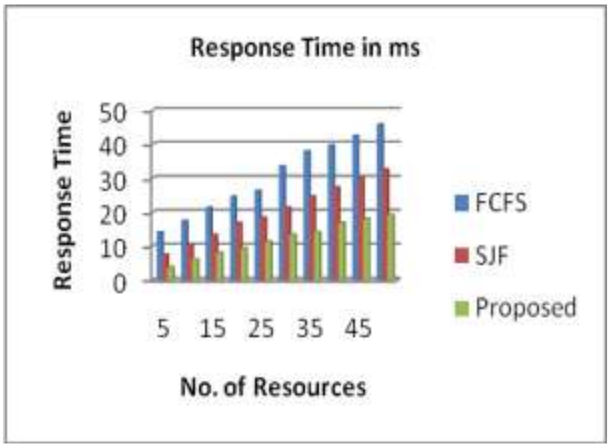

Figure 5: Response Time in MS

The Graph shown below is the Turn Around time to access the resources for the users of the cloud. Here comparison between First Come First Server, Shortest Job First and Proposed techniques for various resources is done.

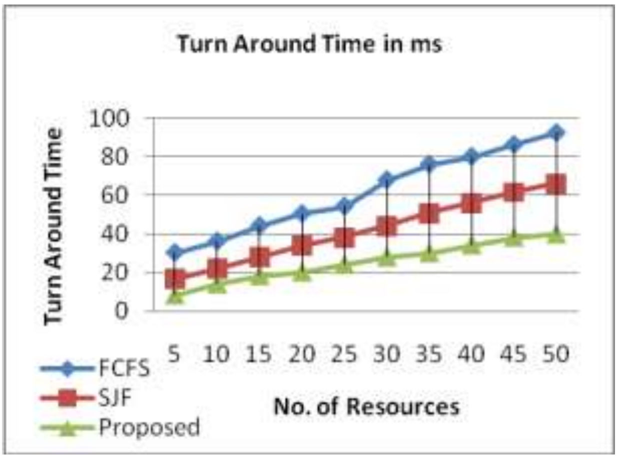

Figure 6: Turn Around Time in MS

10. CONCLUSION AND FUTURE WORK

The proposed methodology implemented here provides high rate of scheduling and less waiting time for the resources to access. The proposed methodology uses the hybrid combination of shortest job first and priority based scheduling algorithms to provide high computation and provides scheduling of resources in quick time.

The result analysis shows the performance of the proposed methodology. A Simulation is done on various scheduling technique, but the proposed methodology provides efficient waiting time and turnaround time.

The proposed methodology when implemented on the cloud simulation environment provides only about 300.06 overall average response time in the cloud simulation environment.

While in terms of computational cost will only provide about 0.57$.

When comparison is done between First Come First Server Scheduling and Shortest Job First Scheduling and proposed scheduling, the proposed methodology provides less processing time among all.

When comparison is done between First Come First Server Scheduling and Shortest Job First Scheduling and proposed scheduling, the proposed methodology provides less waiting time as compared to other, it is tested on 5,10,15,20,25,30,35,40,45,50 resources in which the proposed methodology provides waiting time less.

When comparison is done between First Come First Server Scheduling and Shortest Job First Scheduling and proposed scheduling, the proposed methodology provides less Response time among all.

When comparison is done between First Come First Server Scheduling and Shortest Job First Scheduling and proposed scheduling, the proposed methodology provides less turnaround time among all.

REFERENCES:

[1] Jianfeng Zhao, Wenhua Zeng, Miu Liu, Guangming Li, "A model of Virtual Resource Scheduling in Cloud Computing and Its Solution using EDAs", JDCTA: International Journal of Digital Content Technology and its Applications, Vol.

6, No. 4, pp. 102-113, 2012.

[2] LIANG Quan, HE Wen-wu, LONG Jian-hui, "A Real-time Service Model Based on Cloud Computing in the Next Generation Internet", IJACT: International Journal of Advancements in Computing Technology, Vol. 4, No. 9, pp. 280-287, 2012.

[3] Guo Xinzhi, Niu Liping, "Research on Risk Assessment Model of Information System Based on Cloud Environment", JCIT: Journal of Convergence Information Technology, Vol. 7, No. 11, pp. 288-296, 2012.

[4] Rui Wang, Qinghua Bai, Jia You, Hualing Liu, "Research on Semantic Metadata Model for Information Resource Management under Cloud Computing Environment", JCIT: Journal of Convergence Information Technology, Vol. 7, No. 21, pp. 18-26, 2012.

[5] Kai-jian Liang, Quan Liang, "A Dynamic Method of Model Constructing For Cloud Computing", AISS: Advances in Information Sciences and Service Sciences, Vol. 4, No. 17, pp. 174-182, 2012.

[6] Wang Juan Li Fei Chen Aidong, "An Improved PSO based Task Scheduling Algorithm for Cloud Storage System", AISS: Advances in Information Sciences and Service Sciences, Vol. 4, No. 18, pp. 465-471, 2012.

[7] Dung-Hai Liang, Peirchyi Lii, Dong-Shong Liang, "Risk Management of Land Use on Cloud Computing", JCIT: Journal of Convergence Information Technology, Vol. 7, No. 1, pp. 122-129, 2012.

[8] A new Class of Priority-based Weighted Fair Scheduling Algorithm, Physics Procedia ,33 (2012) 942 – 948.

[9] Hong Luo, Dejun Mu, Zhiqun Deng, et. Research of Task Scheduling in Grid Computing. Application Research of Computers. 2005; (5): 16 -19.

[10] Xindong You, Guiran Chang, Xueyao Deng, et. Grid Task Scheduling Algorithm Based on Merit Function. Computer Science. 2006; 33(6)

[11] Wen sheng Yao,et al. Genetic Scheduling on Minimal Processing Ele-mentsinthe Grid. Springer VerlagHeidelberg. 2002.

[12] Chunyan Zhao. Research and Implementation of Job Scheduling Algorithm Under Cloud Environment. Master's Thesis of Beijing jiaotong University. 2009.

[13] Yiqiu Fang, Fei Wang, Junwei Ge. A Task Scheduling Algorithm Based on Load Balancing Cloud Computing. Lecture Notes in Computer Science. 2010; 6318: 271-277.

[14] Peng Wang, Jingyi Dong. Research of a realization method of cloud computing architecture. Computer Engineering and Science, 2009.

[15] Zhijia Liu, Tirong Zhang, Xiongchen Xie. Research of "Our customers expect" task scheduling algorithm based on cloud computing. Popular Science, 2011.

[16] Thomas A. Hen zinger, Anmol V. Singh, Vasu Singh, Thomas Wies, "Static Scheduling in Clouds".

[17] T.Casavant and J.Kuhl, "A Taxonomy of Scheduling in General Purpose Distributed Computing Systems", "IEEE Trans. On Software

Engineering", vol.14, no.3, February 1988, pp.141-154

[18] Fatos Xhafa, Ajith Abraham, "Computational models and heuristic methods for Grid scheduling problems", "Future Generation Computer Systems 26", 2010, pp.608 -621.

[19] M.Arora, S.K.Das, R.Biswas, "A Decentralized Scheduling and Load Balancing Algorithm for Heterogeneous Grid Environments", "Proc. Of International Conference on Parallel Processing Workshop (ICPPW'02)", Vancouver, British Columbia Canada, August 2002, pp.400-505

[20] Yun-Han Lee et al, Improving Job Scheduling Algorithms in a Grid Environment, Future Generation Computer Systems, 27 (2011) 991–998.

[21] Eucalyptus Cloud Computing Software. http://www.eucalyptus.com/.

[22] Open Nebula. An open source tool kit for data center virtualization. http://opennebula.org/.

[23] Open Stack. Open source software for building private and public clouds. http://openstack.org/.

[24] Wang Juan Li Fei Chen Aidong, "An Improved PSO based Task Scheduling Algorithm for Cloud Storage System", AISS: Advances in Information Sciences and Service Sciences, Vol. 4, No. 18, pp. 465-471, 2012.

[25] Amazon Simple Storage Service. http://aws.amazon.com/s3/.

[26] Shamsollah Ghanbari,Mohamed Othman." A Priority based Job Scheduling Algorithm in Cloud Computing" International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012).

[27] Haiwen Han, Qi Deyu "A Qos Guided task Scheduling Model in cloud computing environment" Fourth International Conference on Emerging Intelligent Data and Web Technologies, 2013.

[28] Mayank Mishra, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo, "Dynamic Resource Management Using Virtual Machine Migrations", Sep 2012, 0163-6804/12, IEEE Communications Magazine, page no: 34-40.

[29] Venkatesa Kumar. V and S. Palaniswami," A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing", 2012, Journal of Computer Science 8 (5), ISSN 1549-3636, Science Publications, page no: 780-788.

[30] Vijindra and Sudhir Shenai. A, "Survey of Scheduling Issues in Cloud Computing", 2012, ICMOC-2012, 1877-7058, Elsevier Ltd, Doi: 10.1016/j. proeng.2012.06.337, page no: 2881 – 2888.

[31] Liang Luo, Wenjun Wu, Dichen Di, Fei Zhang, Yizhou Yan, Yaokuan Mao, "A Resource Scheduling Algorithm of Cloud Computing based on Energy Efficient Optimization Methods", 2012, 978-1-4673-2154-9/12, IEEE.

[32] Xin Lu, Zilong Gu, "A LOAD-ADAPATIVE CLOUD RESOURCE SCHEDULING MODEL BASED ON ANT COLONY ALGORITHM", 2011, 978-1-61284-204-2/11, Proceedings of IEEE CCIS2011, Page no: 296-300.

[33] Zhongni Zheng, Rui Wang, Hai Zhong, Xuejie Zhang, "An Approach for Cloud Resource Scheduling Based on Parallel Genetic Algorithm", 2011, 978-1-61284-840-2/11, IEEE, Page no: 444-447.

[34] Lu Huang, Hai-shan Chen, Ting-ting Hu, "Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing", JOURNAL OF SOFTWARE, VOL.8,NO. 2, FEBRUARY 2013, ACADEMY PUBLISHER,doi:10.4304/jsw.8.2.480-487, Page no: 480-487.

[35] Sunita Bansal et al, Dynamic Task-Scheduling in Grid Computing Using Prioritized Round Robin Algorithm, IJCSI International Journal of Computer Science Issues, 8(2)(2011) 472-477.

[36] Li Yang et al, A new Class of Priority -based Weighted Fair Scheduling Algorithm, Physics Procedia, 33 (2012) 942 – 948.

[37] Ghanbari, Shamsollah, and Mohamed Othman. "A Priority based Job Scheduling Algorithm in Cloud Computing." Procedia Engineering 50 (2012): 778-785.

[38] T.L. Saaty, The Analytic Hierarchy Process,(New York 1980).

[39] Syed Nasir Mehmood Shah, "Agent Based Priority Heuristic for Job Scheduling on Computational Grids" Proceedings of the International Conference on Computational Science, ICCS 2012.

[40] Haruna Ahmed Abba Design, Development and Performance Analysis of Deadline Based Priority Heuristic for Job Scheduling on a Grid, High Performance Computing Centre, Universiti Teknologi Petronas, 31750, Tronoh, Malaysia Procedia Engineering 50:397–405. DOI: 10.1016/j.proeng.2012.10.045

[41] Hédi Chtourou, "A two-stage-priority -rule-based algorithm for robust resource-constrained project scheduling, Computers & Industrial Engineering Volume 55, Issue 1, August 2008, Pages 183–194.

[42] S. Ferretti, V. Ghini, F. Panzieri, M. Pellegrini, and E. Turrini. QoS-aware clouds. In 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD), pages 321–328, Jul. 2010.

[43] R. N. Calheiros, R. Ranjan, A. Beloglazov, et al. "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms". Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Australia.