CHAPTER 6
RESERVATION AND ON-DEMAND PRIORITY BASED QUEUE JOB SCHEDULING ALGORITHM

6.1 INTRODUCTION

The previous chapter deals with TQS algorithm for reservation phase of cloud computing. It ignores to deals with on-demand phase of cloud computing. The proposed research work furnishes more significance to Reservation and On-demand Priority (ROP) phase in addition to that existing time constraint based. To evade this fragmentation needs a ROP based queue job scheduling algorithm is essential for cloud computing. The proposed priority based queuing is used to overcomes the problem posed by the existing system. Based on the log register the frequently used jobs have more priority and least used jobs are scheduled in low priority. Competence of resource allocation depends upon the dynamic metascheduler. To formulate the well-organized use of the resources, the anticipated scheduling algorithm achieves the optimization for cloud scheduling problems.

In Shortest Job Backfilled First (SJBF) used the system-predicted time [20] is better one when compared to user-estimated runtime. FCFS is the easiest, expedient, fair and predictable scheduling. Major disadvantages of FCFS is the average waiting time of job which is very long, poor resources usage leads to fragmentation and low down in utilization rate. The various backfilling policies like conservative, aggressive backfilling are enhanced with FCFS. The already existing backfilling policies are enhanced by either conservative backfilling or aggressive backfilling. It is called as EASY backfilling. These policies enhance higher resource usage and diminish the average waiting time of jobs at some level. CBA [82] is able to picks a variety of different small jobs to scheduling on appropriate idle resources. It can backfill different jobs at one single time to increase the utilization of resources in the system.

The mechanism in backfilling scheduling is under estimation of job while at the time of running. These under-estimated jobs are continuing their execution after the timeout. These jobs are waited by other jobs that are being executing with
advanced reservation. A feasible backfilling policy [111] is needed for scheduling of jobs on diverse type of jobs. Based on the flexible backfilling strategy, jobs can be scheduled to execute according to the priority of jobs submitted by users. The deadline, execution and queue waiting time is predicted with the help of priority value calculated by heuristic method. In CBA the state of the unused resources is searched by suitable jobs from the queue, it picks a group of small jobs to backfill the resources gap to increase the usage of resources.

6.2 ROP JOB SCHEDULING ALGORITHM

The reservation phase has high priority one and jobs are already reserved for accessing their service from cloud service provider. For that reason the Service provider first allocate the server to reservation phase. If scheduler assigns high priority to the on-demand user, the registered customers lead to dissatisfaction about the service provider. In every aspect the service provider give more importance to already existing customer rather than new customer. The scheduler assigns least priority for On-Demand phase. In the second stage of the ROP queue jobs are in ascending order based on their execution time. The jobs are divided into small, medium and large based on execution time.

The following Figure 6.1 shows the Architecture of ROP based Queue job scheduling for Cloud Computing. The customer submits their jobs and ask request to the service provider. Cloud Computing consists of two stage process. In the first stage is reservation and second phase is on-demand. The queue manager collects the information about the user submitted jobs on the basis of arrival sequence and updates them. The primary function of global scheduler is liable for assigning the suitable jobs to the processor. The queue manager divides the queue into reservation and on-demand. The reservation queue has the most priority and on-demand queue in the least priority. The scheduler arranges the jobs in ascending order. The priorities of jobs are used to create a quality system. The queue manager is responsible for handling the queue and allocates the job id for each job.
The process of CBA [109] for parallel job scheduling is as follows:

The newly arrived jobs are submitted to the scheduler. There are three state of resources like the running job is finished, the job fails to run or the job terminated by client.

Step 1 Get the state of resources in computer node, estimate the number of idle resource nodes and their free time span. Call function GetIdleResources().
Step 2 The starved idle resources are search by a grouping of jobs to be backfilled from the waiting jobs queue. Call function GetComJobs().

Step 3 The grouping of jobs given in step (ii) is scheduled to execution on the suitable idle resources.

Calculate the list of idle computing nodes:
Function GetIdleResources()

(i) picks the current free computing nodes,
(ii) estimate the idle time span for each computing node,
(iii) sort the computing nodes in ascending order in accordance with the idle time span given in step (ii),

The list of the idle computing resources in the number of idle computing nodes be ‘X’
}

Function GetComJobs()

The list of the idle computing resource nodes from function GetIdleResources(),

Estimate ‘X’-Nidle, find the start time S of the job at the head of the waiting queue.

Let ‘S’ be the starting time of the job, in addition to the idle computing resource nodes for the first job in the queue. If there are new added idle computing nodes arrived, the number of the remaining nodes become called as ‘Ne’.

(i) Small Jobs =the jobs between 1&40% from the sorted job list.
(ii) Medium Jobs =the jobs between 41&80% from the sorted job list.
(iii) Large Jobs =the jobs between 81&100% from the sorted job list.

First gives more preference for small jobs because the execution time is very low and take more jobs to finish with in short duration. Next the assign priority for medium jobs that takes more time when compared to small jobs. Finally the large jobs take more time for execution so the scheduler allocates 20% from the total
submitted sorted jobs. Queue manager process the jobs based on burst time, successful completion jobs are enter into the cloud computing. The service requested by the customers, scheduler reduces the waiting time of the task and maximize the QoS.

6.3 RESULTS AND DISCUSSION

The main objective of research work provides QoS. The less time taken for completion jobs decides the effective scheduler. The ROP job scheduling algorithm is compared with the existing scheduling algorithms like traditional backfill, combinational backfill and MQS. ROP produced an optimum solution for job scheduling in cloud computing.

6.4 COMPARISON OF ROP WITH TRADITIONAL BACKFILL ALGORITHM

![Figure 6.2 Performance of ROP based queue with Traditional Backfill](image)
The above Figure 6.2 shows the performance of ROP based queue with Traditional Backfill. The X-axis represents the number of cloudlet and Y-axis represents the processing time of the job. The traditional algorithm is backfilled with the help of FCFS. The number cloudlet between 2000 – 2700 both these algorithm produce more or less the same. When the size of cloudlet is above 2700, the ROP based queue produces better results when compared to Traditional backfill.

6.5 COMPARISON OF ROP WITH COMBINATIONAL BACKFILL ALGORITHM

The following Figure 6.3 shows the performance of ROP based queue with CBA. The X-axis denotes the number of cloudlets and Y-axis denotes the processing time.

![Job Completion Ratio](image)

**Figure 6.3 Performance of ROP based queue with CBA**

The job completion ratio in terms of number of cloudlets between 0-750 both methods same results. In overall ROP based queue provides better results when
compared to CBA. Using CloudSim, the simulated results are obtained for various types of jobs in the Cloudlet.

6.6 COMPARISON OF ROP WITH MQS ALGORITHM

The following Figure 6.4 shows the performance of ROP based queue is compared with MQS. The cloudlets are represented in X-axis and processing time is represented in the Y-axis.

![Figure 6.4 Performance of ROP based queue with MQS](image)

The number of cloudlets below 2600 the MQS takes the advantage of computer nodes usage. If the cloudlet size is above 2600 and above the ROP based queue provides better result.
6.7 COMPARISON OF ROP WITH EXISTING ALGORITHMS

The given below Figure 6.5 shows the performance of ROP based queue compared with Traditional Backfill, CBA and MQS algorithm. The job completion time of the proposed ROP based queue system provides better one when compared to the existing FCFS, CBA and MQS job scheduling algorithms.

![Graph showing Comparison of ROP with Existing Algorithms](image)

Figure 6.5 Performance of ROP based queue with Existing Algorithms

6.8 SUMMARY

This chapter describes the brief overview of the ROP based queue job scheduling algorithm. The queue manager splits separate scheduler for on-demand and reservation phases. It gives beneficial for both reservation and on-demand phases. The proposed ROP based queue algorithm gives importance to execute the jobs having high priority job in the reservation phase queue. Next it executes on-demand phase jobs having least priority. The important factor that affects the performance of the system based on minimum makespan, optimum load balancing and maximum resource usage.