ABSTRACT

The buzzword in the present computing world, cloud computing has emerged as a lucrative business model for the utility computing system. All around the globe, the interest towards the cloud computing is surging. It has successfully enabled scalable and virtualised infrastructure that has put extensive computing abilities in the hands of consumers. Through virtualisation, multiple users are able to access the computing resources at any given time as the software; data and resources are virtually created and delivered. Cloud computing offers all the computational resources and services via internet. Cloud computing can also be defined as an exceptional platform with a huge potential that can host a large number of computing services that continuously change over time due to random submission of jobs by the users in a large number. Thus, the computing services always vary according the user requirements.

Cloud computing ensures scalable, pay-per-use, reliable, customised and dynamic computing environments for the end users. Cloud computing, as a vastly emerging technology, is also leading to the rise in user requirements. This ideal model of cloud computing is appealing to many vendors and associations who have just begun to understand the profits they can reap by making use of cloud computing. Shifting their data and applications to the cloud will make them big profits as this paradigm is productive and not too cosltly. Moreover, it can handle large computational problems in a much capable and easy manner. On the other hand, the main focus of the cloud providers is on management of computing power, storage, energy consumption and offering required services to the users via internet. Managing the user requirements on such a large and varied scale has also created several challenges in spheres of job scheduling and on-demand resource allocation.

Cloud infrastructure is largely responsible for bringing the need of efficient job scheduling systems and framework to the front. Scheduling problems have always been the core challenging issue in the computing systems. It is quite challenging to manage and schedule the jobs originated from different cloud users to the available resources in a performance-optimised manner. Thus, several job scheduling algorithms have been designed that vary in design and behaviour and attempt to handle various issues like user-share fairness, resource awareness, energy saving and

locality of data. These scheduling algorithms are important because it is crucial to schedule the cloud user jobs in an optimised manner. It is also essential that a job scheduler picks an appropriate algorithm, with the aim to maximise performance, which is a very challenging hypothetic issue in the cloud computing domain.

The proficiency of the scheduler has a great impact on determining the adequacy of the cloud environment. The scheduler manages all the jobs, and machines and performs several tasks like prediction of the incoming workloads, provisioning of the necessary VMs in advance, allocation of the jobs to the VM, releasing the idle VMs, starting necessary VMs by allocating jobs for the execution. It is up to the cloud service provider to implement the scheduler, that will produce the most optimal job scheduling through proper distribution and execution of jobs on the available set of resources while maintaining the cost and energy efficiency and reducing the execution time as much as possible; thereby strengthen the performance.

Multiple jobs processing is done on multiple machines. All of these works goes in parallel. It is true that the traditional scheduling algorithms work great but in theory. They struggle when multiple machines are brought into the picture. With the rise in cloud capabilities, user requirements and users, traditional algorithms are bound to fail, as presented in the earlier research. The problem with the algorithms like First Come First Serve, Round Robin, and Shortest Job First scheduling is that the process doesn't terminate in the scheduled time, which in turn elongates the waiting time for other jobs in the queue. Due to this, fragmentation occur leading to wastage of energy and surging costs of the customer on the pay-per-use model.

In the earlier research, an efficient multi-queue job scheduling known by the name Efficient Multi Queue Scheduling (EMQS) has been proposed in which the client job is categorised into three job queues, small, large & medium, and then, the job selection is performed dynamically to reduce the waiting and processing time for any process in the queue. Therefore, it solves the fragmentation issue that is associated with the traditional job scheduling algorithms. The work done in this thesis analysed and implemented this approach and we noted that it gave importance to all the jobs in a fair manner. However, managing three queues gets hectic as the scheduler needs to switch from one queue to another way too often, thereby affecting some ready jobs that have to keep waiting in the queue. Also, scheduler carry out dynamic selection of

user jobs that can also lead to indefinite postponement of some processes, and miss their deadlines.

Keeping the performance parameters such as energy saving; in this thesis work, inspired from the concept of multi-queue scheduling algorithms proposed an algorithm recognized by the name smarter multi-queue scheduling (SMQS) algorithm for scheduling cloud user jobs. The said algorithm accommodates the virtualisation trait of the cloud computing and is superior to the traditional job scheduling algorithms as it concentrates on the energy efficiency while maintaining the resource fairness by doing optimal jobs allocation.

The contributions of our work are summarised below:

1. Firstly, the research work in this thesis presented insight study about cloud paradigm focusing on existing job scheduling algorithms available in cloud environment for scheduling cloud user jobs. Then out of these studied algorithms; two heuristic based job scheduling algorithms - Ant Colony Optimization (ACO) and Efficient Multi Queue Scheduling (EMQS) Algorithms are selected and implemented to schedule the cloud user jobs. The simulation results showed the efficiency of the Efficient Multi Queue Scheduling (EMQS) Algorithm over Ant Colony Optimization (ACO) algorithm in terms of energy consumption and time consumption.

2. Next, the research work proposed the framework model to manage the cloud user jobs and resources in the cloud. The framework recognizes by the name Smarter Multi Queue Scheduling Algorithm. The framework of this algorithm constitutes of four components - Scheduler, Queue manager, MPI table and Network. The Scheduler forms and runs an optimal scheduling strategy by using the other major components. The queue manager classifies and manages the user jobs whereas the network contains the cluster of virtual systems. The MPI Table helps to collection the information about the vacant systems in the network and maintains this information in a sorted list.

3. Carrying research work forward, sucessfull implemention of proposed Smarter Multi Queue Scheduling Algorithm is achievd. The algorithm works in manner that jobs qualification is performed by establishing two job queues - small and large. A threshold prioritised value is calculated and the jobs get allocated to both the job

queues based upon this calculated value. Then we analysed and executed a distribution strategy for the user submitted jobs. The user process from both the queues are picked and they form a series of merge task sets with each set containing two merge tasks (one from small queue and one from the large one). Later, in a sequence, the scheduler allocates these unmapped merged processes to the available resources in an optimised manner.

4. In the final step of research work, efforts have been made to evaluate the performance of our said Smarter Multi Queue Scheduling Algorithm (SMQS). The simulation is achieved in the cloud environment to find the best schedule to process the cloud user jobs on the available virtual machines.The overall performance of said proposed algorithm is examined and measured by running it under randomly generated cloud configurations. The simulation was carried in real-time environment. For example, n different user jobs with different characteristics are randomly created. The simulation environment is provided by Microsoft Windows Azure platform using SQL which offers runtime execution environment for the managed code to host and run scalable solutions using the Microsoft .NET framework. Experimental results deduced that when implemented and compared, proposed Smarter Multi Queue Scheduling (SMQS) Algorithm outperforms the existing Efficient Multi Queue Scheduling (EMQS) Algorithm on the parameters of energy consumption and time consumption.