CHAPTER 7

CONCLUSION

This chapter presents the contributions and critical analysis of this thesis work. Suggestions for future work have also been outlined. Before proceeding with the review of the work done, the objectives stated earlier in the introductory chapter are recalled.

The primary objective of this thesis work is a workflow scheduling methodology that aims to optimize resource utilization in Cloud Computing. WFS-ABC aims at efficient provisioning of the cloud resources while considering multiple QoS parameters: completion time; cost; and completion deadline. WFS-ABC takes into consideration the precedence relationship among the various tasks while scheduling the tasks so as to meet the completion deadline. The methodology, in addition, looks at balancing the task load among the multiple VMs of the cloud computing resources. The research work has used ABC algorithm for the resource optimization process.

7.1 CONTRIBUTIONS OF THE THESIS

The research work has proposed a workflow scheduling methodology, WFS-ABC, for optimizing the resource utilization in Cloud Computing. Service providers typically enter into an agreement with the end consumers for providing cloud utility services. The terms and conditions of delivering the service are documented in a SLA. Such terms would include cost, application completion time and other QoS criteria which would have to
be met by the service provided to the consumers. More often than not, penalty clauses are included in the SLA to define the financial implications like fines or forfeits, when the services provided to the consumer fails to meet the terms and conditions.

WFS-ABC workflow scheduling has been developed to schedule workflows so as to meet the multiple criteria defined in a SLA. The QoS criteria considered for this thesis work are execution time, cost and deadline. Additionally, load balancing is considered to divide the task load in a manner that would balance the task load among the multiple VMs of the cloud environment.

It should be mentioned that prior work on workflow scheduling in cloud computing has considered only one or two QoS parameter. For instance, GA has been used for optimizing the completion time, while ACO has been used for optimizing load balancing. PSO had been proposed for optimizing cost and completion time. Hence, it can be seen that prior work has not considered more than two criteria for optimizing workflow scheduling.

This thesis work has first looked at workflow scheduling while minimizing the single QoS criteria of task execution time. The work has been extended to consider two QoS criteria of execution time and cost. The next step has been to consider multiple QoS criteria of execution time, cost and deadline. The final work has been to balance the load among the multiple VMs in the cloud while scheduling workflows that considers all the three QoS criteria, so as to arrive at an efficient utilization of cloud resources.

This thesis work has used ABC algorithm for optimizing the resource usage. ABC algorithm is used extensively for solving multidimensional and multimodal optimization problems and has the
advantages of strong robustness, fast convergence and high flexibility with fewer control parameters (Karaboga 2005). In comparison, with the other methods, GA, ACO and PSO discussed in Chapter 2, ABC has only one control parameter of maximum cycle number (Karaboga & Akay 2009). GA has three control parameters of crossover rate, mutation rate and generation gap (YujiaGe 2010), while ACO has three control parameters, pheromone, relative weight and visibility. ACO has poor converge and stagnation problem when all jobs require or assigned to the same resources (Alyaseri 2013). Likewise, PSO has three control parameters cognitive, social factors and inertia weight (Kennedy & Eberhart 1995). Hence, ABC has a faster convergence rate and the optimized result can be arrived at relatively faster.

In the first work, WFS-ABC methodology was developed to minimize the job completion time in scheduling the workflows while keeping in mind the task dependencies. The initial solution was framed using FIFO, with each task having a time quantum for completion of that task. The performance of WFS-ABC was compared with the other methods GA, PSO and ACO methods. As each of these methods have looked at different criteria, to enable performance comparison, GA, PSO and ACO methods have been applied for execution time criteria and the performance was compared. For example, the results show that there is a reduction in execution time with WFS-ABC as compared to the other algorithms. For instance, considering a task count of 100 tasks, there is a reduction of 16 % in time when compared with GA, a reduction of 11 % as compared with ACO and 5 % as compared with PSO. This shows that WFS-ABC performs better than GA, ACO and PSO for single criteria.

In the second work, cost and completion time were considered for scheduling of the workflows. A priority assignment for managing the large number of requests generated by cloud users was devised to arrive at a
specific order for the execution of the tasks. The workflow scheduling was developed to minimize cost as well as job completion time while maintaining the defined order of task execution. The performance of WFS-ABC was compared with GA, PSO and ACO. In this instance as well, the results show that there is a reduction in execution time with WFS-ABC as compared to the other algorithms. For instance, considering a task count of 100 tasks, there is a reduction of 12 % in time when compared with GA, a reduction of 7 % as compared with ACO and 3 % as compared with PSO. This shows that WFS-ABC performs better than GA, ACO and PSO for multiple criteria.

The next work was to extend WFS-ABC to satisfy the multi criteria QoS requirements of minimizing cost and completion time while maintaining the overall task completion deadline. Tasks can be completed earlier using more expensive services but child tasks cannot be started till all its parent tasks have been completed. Hence task dependencies have been considered while scheduling tasks in parallel, and minimization of execution times as well as cost is both considered during task assignment. Once again the performance of WFS-ABC was superior as compared to the other algorithms. For a task count of 100 tasks, there is a reduction of 14 % in time when compared with GA, a reduction of 8 % as compared with ACO and 4 % as compared with PSO.

For efficient utilization of cloud resources, it is essential that the load is balanced evenly across the multiple VMs used for task execution. For instance, if tasks are assigned only to VMs, which have a better configuration, there could be heavy load on certain VMs while others are idle leading to resource wastage. This may also lead to increase in cost. This work has therefore looked at balancing the load along with minimization of overall task completion time and cost while maintain the task dependencies and completing task within the permitted completion deadline. The performance
results show that WFS-ABC has the optimum utilization of VMs and also a better fitness value for the combined multiple QoS criteria. Considering the same count of 100 tasks, there is a reduction of 10 % in time when compared with GA, a reduction of 6 % as compared with ACO and 2 % as compared with PSO. In addition, the increase in completion time is sharp up to a task count of 75 and thereafter, the increase is minimal and the graph shows that the execution time curve tends to become flat. It can be observed that WFS-ABC has minimum completion time. It should be noted that performance of WFS-ABC which considers multiple criteria outperforms over the existing algorithms by 5% which consider only a single criteria.

The workflow scheduling methodology proposed in this thesis can be effectively used in emerging services of SaaS and XaaS models of cloud computing. Services running in the cloud can be scheduled to effectively meet the SLA terms and conditions with respect to multiple criteria, like, execution time, cost etc.

7.2 CRITICAL ANALYSIS

WFS-ABC scheduling methodology is used to schedule workflows based on multiple criteria. This can result in a trade-off among the different criteria which are likely to conflict with other. For example, cost is an important criterion which should be kept low. If only cost is considered, then the cost achieved may be much lower than that obtained by looking at minimization of cost along with minimization of other criteria such as execution time and load balance.

For instance, while considering the performance for a task count of 100, percentage reduction of WFS-ABC with respect to GA is 16% considering only one criteria, namely, completion time. However, when the two criteria of completion time and cost are considered, then the percentage
reduction comes down to 12% and when all criteria and load balance is also considered this reduces to 10%. The same is found to be true with respect to ACO and PSO. Hence, multiple criteria would lead to overall efficient resource utilization but may not achieve the lowest value for each criterion, as compared to considering each criterion individually.

This research work has looked at the multiple criteria of execution time, cost, overall task deadline and load balance. Other criteria such as user defined priority, availability, reliability can also be considered as future work.

7.3 FUTURE WORK

Intelligent Water Drops (IWD) algorithm (Hosseini 2008) is an emerging swarm-based nature-inspired optimization algorithm. IWD is a population-based algorithm where each drop represents a solution and the sharing between the drops during the search lead to a better drops (or solutions). There is a need to explore the application of such emerging algorithms for cloud resource optimization (Niu et al 2012).

The cloud computing paradigm is still vulnerable to a large number of system failures and, as a consequence, there is an increasing concern among users regarding the reliability and availability of cloud computing services. There is also an increased requirement for fault tolerance to achieve reliability for real time computing on cloud infrastructure (Nimisha & Seema 2013). There are many challenges when it comes to implementing these concepts to cloud computing. In most of the real time cloud applications, processing on computing nodes is done remotely. So there are more chances of errors. The fault tolerance model should aim to take advantage of the distributed computing capacity that is available and the scalable virtualized environment of cloud computing for better implement of real time
application. The system should be able to tolerate the fault proactively and makes the diction on the basis of reliability of the processing nodes.

Check pointing is the traditional method of fault tolerance that is used for long running and big applications. When a task fails, rather than commencing execution from the beginning, jobs can restart from the recently checked pointed state. However, in cloud computing, the user is not aware of where the job is executed and because of VM migration; the entire job may not be executed on the same VM. So it becomes necessary to implement the checkpoint across multiple virtual VMs as well as multiple physical servers. This remains a big challenge and can be taken up as future work.

Cloud Computing has become an increasingly popular means of delivering valuable, IT enabled business services. User requirement for compute time has to be dealt with on a continual basis. The jobs have to be allotted to the appropriate cloud nodes keeping in mind the QoS requirements. In this scenario, there is need for an efficient failure handling strategy to restart and resume failed jobs. Future research work can be focused on availability for cloud resources.

The most important cause of concern, however, is related to security, privacy, and trust. Security in particular, affects the entire Cloud computing stack (Pearson & Benameur 2010). The Cloud computing model promotes massive use of third party services and infrastructures to host important data or to perform critical operations. In this scenario, the trust towards providers is fundamental to ensure the desired level of privacy for applications hosted in the Cloud. Thus the possibility of exposure and sharing of data should be taken into account while scheduling tasks and data on Cloud resources. The challenge is the introduction of these constraints into the scheduling of data and tasks while mapping tasks to resources in the Cloud.
Data centers are expensive to operate as they consume huge amount of power. The carbon footprint on the environment is rapidly increasing. In order to address these issues, energy efficient resource allocation and algorithms need to be developed. The performance of data centers depends on the provisioning and usage of its hardware devices by the VM management software depending on user needs. As more CPUs are used, the temperature of the hardware increases. This requires cooling of the data center. Hence, data centre performance and energy consumption are directly related to each other. As the price for commodity hardware and network equipment’s for a data center is already getting cheaper, significant part of the total cost of operating Cloud services in industrial scale is determined by the amount of energy consumed by the data center. To conserve energy and save cooling costs, data centers could adopt energy efficient scheduling policies. Application schedulers could make use of statistical information obtained from sensors when submitting tasks and data for computation across VMs.

7.4 SUMMARY

The WFS-ABC methodology addresses the key challenges relating to workflow scheduling. The focus has been to efficiently schedule workflows so as to satisfy the multiple QoS requirements of customers.

The WFS-ABC scheduling methodology takes into consideration the precedence relationship among the various tasks and completes all tasks within their permitted deadline. The methodology also looked at balancing the task load among the different VMs. The workflow scheduling methodology, therefore, provides an efficient provisioning of the cloud resources while minimizing completion time and cost; meeting the task deadline as well as balancing the load on resources used.