American Journal of Engineering and Applied Sciences

Original Research Paper

Scheduling Strategies in Cloud Computing: Methods and Implementations

Aida Abou-Elseoud Nasr and Sheren Ahmed Elbooz

Department of Computer Science and Engineering,
Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt

Article history
Received: 18-07-2017
Revised: 20-03-2018
Accepted: 04-4-2018

Corresponding Author:
Aida Abou-Elseoud Nasr
Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf 32952, Egypt
Email: aida.nasr2009@gmail.com

Abstract: A Cloud computing allows to users to get computing resources and services over the internet in the form of an on-demand service using virtualization technology. While cloud resources are built in different locations so they need new strategies to manage the running of them. These strategies may contain more than one algorithm. In cloud computing system, scheduling system is the one of the main keys of system's performance. The scheduling system doesn't include task scheduling strategies only, but also it contains fault tolerance and load balancing strategies. Scheduling system is used in cloud management and handling failure. In this study, we discuss some scheduling algorithms and their advantages and disadvantages. Also we guide readers to choose the best for achieving the high quality.

Keywords: Scheduling Algorithm, Cloud Computing, Load Balancing, Fault Tolerance

Introduction

Cloud computing system has pool of resources for the users' applications execution. These resources often appear to be unlimited. Google, Microsoft, Amazon and other companies use cloud computing to deliver their services. The simplest discretion of cloud computing is "cloud computing is efficiently saving and accessing data and applications over the Internet from any location by using computer, smart phone or lab top". Most of the IT institutes have explained cloud system as a system for on demand web based execution of a pool of virtual resources which the most attractive issue of these resources is elasticity (i.e., multiple users can use multi-rent model with different allocating and reallocating of physical and virtualized resources according to users' needs) (Chandrasekaran, 2014).

Four models are implemented in cloud: Private, community, public and hybrid cloud Fig. 1 illustrates the cloud deployment models. Private cloud is purveyed for special use by a one organization containing multiple clients. On the other hand, public cloud is purveyed for many clients. It is open use cloud. The third model is community cloud. In this model, different companies are sharing the infrastructure of a specific community where some interests are shared. Finally, the hybrid cloud is an installation of two or more various cloud models (public and private, public and community, private and community, or public and private).

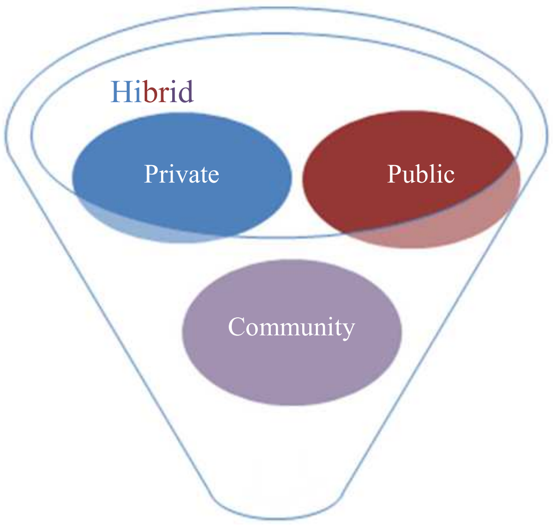

Fig. 1: Cloud computing models

Cloud providers can provide three services in the cloud: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) Fig. 2 illustrates these services.

Using cloud computing services depends on pay-per-use model. The various cloud providers have different schemes which vary in price and computing power.

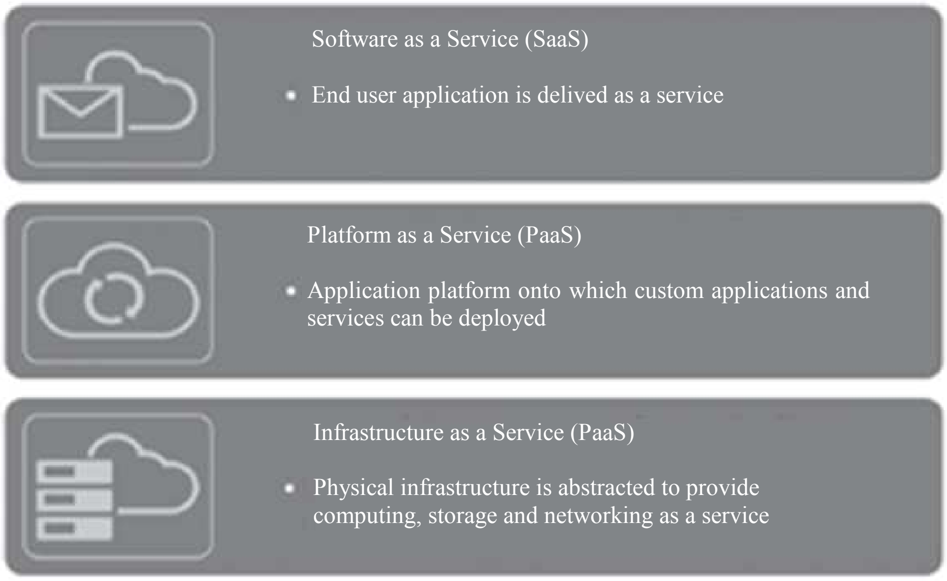
Software as a Service (SaaS)
End user application is delived as a service
Platform as a Service (PaaS)
Application platform onto which custom applications and services can be deployed
Infrastructure as a Service (PaaS)
Physical infrastructure is abstracted to provide computing, storage and networking as a service

Fig. 2: SPI-service offering model of the cloud (Chandrasekaran, 2014)

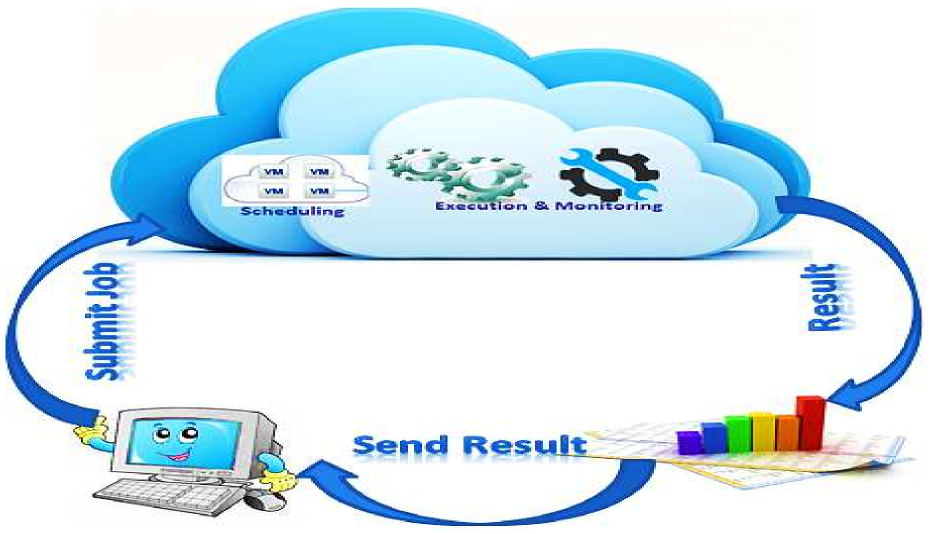

Fig. 3: Job lifecycle on cloud computing

For example, Amazon EC2 has different space of prices model in an On-Demand instances. While Google Compute Engine charges for usage is monthly basis depending on the kind of desired machines. User application passes phases to be processed on cloud system Fig. 3 shows the lifecycle of job execution on cloud computing.

Once the user submits his job to the cloud, scheduling system must schedule and monitor tasks execution. The scheduler cares about minimizing total execution time and the monitory cost. So cloud environment must also takes into account monitoring job execution and sends the correct result to the user. In this study, I tackle with the most important challenges, such as scheduling problem, load balancing and faulty of tasks, that face job execution on cloud system and the recent techniques for solving those problems. Improving system performance depends on improving scheduling system. Scheduling system is not only scheduling algorithm but also include another algorithms to enhance system consumption. All of algorithms are called scheduling system. We will introduce them in the next sections in our paper. In this study, we illustrate scheduling algorithms on cloud computing environment. Also, we take into account the advantages and disadvantages.

Task Scheduling Algorithm on Cloud Computing

In cloud environment, the efficient and powerfully scheduling algorithm must have an essential role in minimizing total execution time and the monitory cost. Despite of the widely distribution of the cloud computing resources offered for cloud users to achieve their needs, it is very important to choose the efficient scheduling strategy to adapt to different business requirements. The cloud owners and managers must define the scheduling strategies to manage their resources and satisfy user needs. Each company selects the scheduling algorithm that satisfies its strategy requirements. For example, Amazon's Cloud scheduling methods mix methods for cost first and then meet different user needs such as load balancing or high reliability. HP company also began data center research work and focuses on data center cost model. On the other hand, the VMW are Company focuses on resource virtualization, disaster recovery and dynamic migration.

Task scheduling algorithms are important to manage the resource allocation. Scheduling is a mapping process of tasks into the available resources (Aida *et al*., 2014). In this study, we explain two types of scheduling methods, heuristic scheduling (Aida *et al*., 2015b) and meta-heuristic (Wenhong and Zhao, 2015). The different types of scheduling methods are created to achieve the near-optimal solution of the resources schedule efficiency at lower time (Aida *et al*., 2016).

Heuristic Method

Researchers are using heuristic methods to get a near optimal solution of a scheduling problem depending on their heuristics (Aida *et al*., 2015a). Each heuristic algorithm uses a group of specific rules to increase the probability of getting a good solution in low time complexity. In task scheduling problem domain, there are three heuristic-based methods classes: List Scheduling and Cluster-Based and Duplication-based Scheduling. The popular type of heuristic scheduling is a list scheduling algorithm. The list scheduling algorithm consists of two phases: A task prioritizing phase and virtual machine selection. In task prioritizing, the task priority is computed firstly. In the other phase, each task (in order of its priority) is scheduled to virtual machine that minimizes a suitable cost function. List-scheduling is accepted as an efficient approach, because it finds a solution in low complexity and with good results. As an example for list scheduling algorithms, George and Beena (2015), developed CFCSC list scheduling algorithm. This algorithm consists of two stages. The first stage sorts tasks in the descending order using upward rank (Ranku) for prioritization. At the same time, the available resources are ordered in an ascending order based on the costs. As one of the objectives is to minimize the monetary cost, the virtual machine with the least cost is given the highest priority. This is implemented for load balancing and to minimize the total price, which includes only the virtual machine cost. As a final step the algorithm calculates the Modified Earliest Finish Time (MEFT) and selects the virtual machine with minimum MEFT value, preserving the precedence constraint. Another heuristic algorithm, Verma and Kaushal (2015) presented Budget and Deadline Constraint Heterogeneous Earliest Finish Time (BDHEFT). Like CFCSC algorithm, the BDHEFT algorithm is based upon HEFT algorithm, which minimizes the overall execution time of a workflow without considering the monetary cost and budget constraints while making. Paper (Bossche *et al*., 2013) proposed a set of scheduling methods to schedule the deadline constrained bag of tasks applications on hybrid clouds to minimize the execution cost. In addition, these algorithms are only suitable for independent tasks.

Guided Random Search Based Method

Guided random search based technique or (meta-heuristic method) finds space of solutions and returns the best one as a result. It takes high time, but it is more efficient and can achieve good performance. A well-known example of guided random search techniques includes Particle Swarm Optimization (Venter and Venter, 2002), Genetic Algorithms (GA) (Haupt and Haupt, 2004), Simulated Annealing (SA), Ant Colony Optimization (ACO) (Dorigo *et al*., 2004) and Tabu Search (TS). Very recently, a new meta-heuristic method, called Chemical Reaction Optimization (CRO) (Xu *et al*., 2010). There are many algorithms that applied the Meta heuristic method. For example, Yuming *et al*. (2013) developed a Double Molecular Structure-based Chemical Reaction Optimization (*DMSCRO*) method for heterogeneous computing systems. In the algorithm, there are two molecular structures: Priority molecule of the tasks in a DAG and molecular of nodes which execute the tasks. The *DMSCRO* method applies the *CRO* algorithms and adds new fitness function which is fit for scheduling DAG. Another example of meta-heuristic methods, Youwei Shao, presented Improved Particle Swarm Optimization (IPSO) algorithm (Shao, 2015), describing an improved PSO algorithm capable to define an optimal solution for the cloud resources. Also, Omara and Arafa (2009) developed two algorithms based on GA technique. The developed algorithms are genetic algorithms with some heuristic principles that have been added to improve the performance. According to the first developed genetic algorithm, two fitness functions have been applied one after the other. The first fitness function is concerned with minimizing the total execution time (schedule length) and the second one is concerned with the load balance satisfaction. The second proposed genetic algorithm depends on task the duplication technique to overcome the communication overhead.

Discussion

With heuristic methods, the time complexity of finding solutions is low, but the quality of this solutions is depend on the effectiveness of heuristics. Although guided-random-search-based scheduling methods typically take longer time, they can achieve good solutions for a wide range of scheduling scenarios. To obtain the high performance, each application should be scheduled by suitable method. In our opinion, scheduler should take into account other sides to manage cloud resources in efficient way. So scheduling system must be interested with fault tolerance to handle system resources and drive job execution in a correct way.

Fault Tolerance in Cloud Computing Systems

Cloud computing is the adaptable technology which provides integration between applications and cloud resources where they are dynamically scalable. In the cloud, customers can access data, applications and platforms using any device over the internet. Due to the large number of cloud resources, the probability of failure becomes high during the execution. So, providers need to fault tolerance approaches to avoid the system failure. There are other benefits of implementing fault tolerance in cloud computing as lowering the cost and failure recovery. In this section we will review some fault tolerance techniques which explain how faults can be solved. Fig. 4 illustrates fault tolerance classification (Bala and Chana, 2012). Many algorithms are developed and used fault tolerance to improve the cloud performance. Kaur and Kinger (2014), (Limam and Belalem, 2014) showed analysis of different techniques used for fault tolerance. In this study, authors illustrated some types of fault tolerance and their different techniques as a survey.

There are many algorithms applied the different fault tolerance techniques, such as, Limam and Belalem (2014) have developed a new method that is a hybrid of tow mechanism: The migration and checkpoint mechanisms. They have developed the checkpoint to reduce the time lost and minimize the effect of the system failures on task execution. The advantage of the second mechanism is ensuring the continuity of task execution and avoiding any loss due to hardware failure. Meshram *et al*. (2013) presented a model for Fault Tolerance in Cloud computing (FTMC) model tolerates the faults on the basis of reliability of each computing node.

Paper (Nazir *et al*., 2009) has used the adaptive check pointing method to tolerate the faults in economy based grid resources. The authors proposed a new adaptive task check pointing based fault tolerant job scheduling method for an economy based grid. The proposed strategy maintains a fault index of grid resources. It updates the fault index dynamically depending on successful or unsuccessful completion of an allocated task. Whenever a grid resource broker has tasks to schedule on grid resources, it makes use of the fault index from the fault tolerant schedule manager in addition to use a time optimization heuristic. While scheduling a grid job on a grid resource, the resource broker uses fault index to apply different intensity of task check pointing (inserting checkpoints in a task at different intervals).

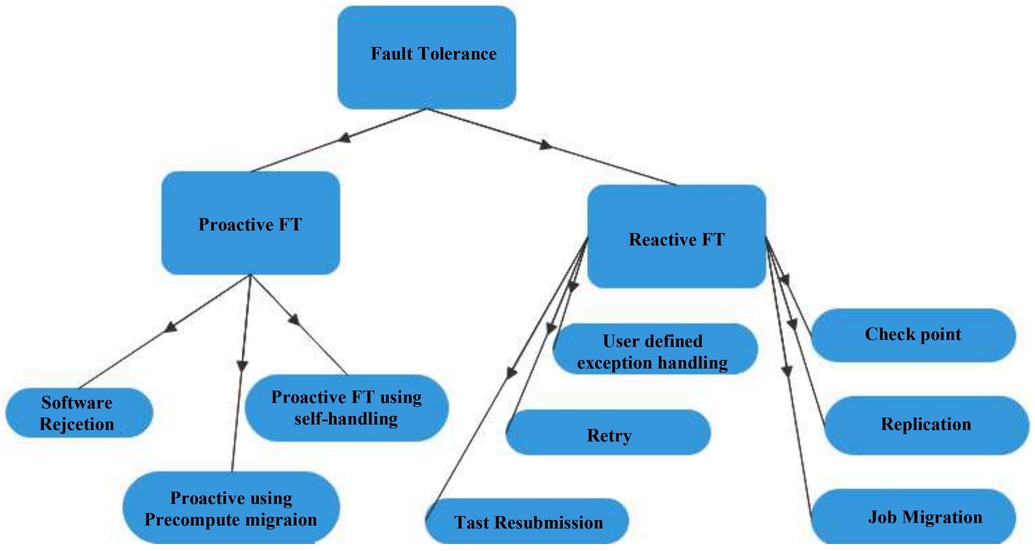

Fig. 4: Fault tolerance techniques

Discussion

Fault tolerance methods play an important role for predicting a system failures and taking an appropriate action before system failures occur. In this study, we will show the different algorithms of fault tolerance and illustrate some of fault tolerance algorithms to improve the cloud performance. However still there are number of issues of implementing fault tolerance technique in cloud computing such as:

- We need to developed autonomic fault tolerance method for multiple instances of an programs running on different virtual resources (Jhawar *et al*., 2013)
- The new fault tolerance method needs to be developed with existing scheduling algorithms
- Some of methods apply replication techniques, this add memory cost
- Check point method takes a long time and adds cost
- Each system needs an appropriate method according to system requirements
- Some of fault tolerance methods need to reboot the system, this take an extra time and add cost

Cloud Resources Load Balance Scheduling Techniques

Cloud data centers are distributed over the internet. Each one contains many nodes such as many servers, storage nodes and network devices. Every node includes series of resources, such as network bandwidth, CPU, memory. The virtualization environment for cloud computing has many advantages, as the ability to migrate an application from resource to another. This ability facilitates manage cloud resources and use the cloud in efficient way. One key of the challenges that plays an important role in cloud system scheduling is resources load balance scheduling algorithms. Load Balancing is the important part of the current virtual environment. Load balancing is the process of redistribution of the load to achieve the high utilization. Fig. 5 shows the meaning of load balance. Many papers have presented load balance scheduling algorithms and they can be divided into two types:

Run Time Load Balance Scheduling Algorithm

With the run time algorithms, scheduler can know the current requests and the statue of virtual machines. A system load may be changed at any time in run time. Authors (Dong *et al*., 2012) proposed a dynamic file migration load balancing algorithm based on distributed architecture. Also Chaczko *et al*. (2011) discussed the load balancing in cloud computing then demonstrates a case study of system availability based on a typical Hospital Database Management solution. In reference (Haryani and Jagli, 2014), the authors presented a new dynamic load balance algorithm checking the counter variable of each server node and data center. After checking, it transferred the load accordingly by choosing the minimum value of the counter variable and the request was handled easily and took a smaller amount of time.

Compile Load Balance Scheduling Algorithm

Here, the schedulers know all requests and every statue of all machines. Compile load balancing methods are non-preemptive i.e., once the load is allocated to the node it cannot be transferred to another node. This method required less communication hence reducing the execution time (Sharma *et al*., 2008). The most famous example for compile load balance algorithm is round robin algorithm (Abubakar and Usman, 2004). In this algorithm, the load is distributed evenly to all nodes. Work load is distributed in round robin order, where equal load is assigned to each node in circular order without any priority and will be back to the first node if the last node has been reached. Each node maintains its load index independent of allocations from remote node. Round robin is easy to implement, simple and starvation free. It does not require inter process communication and gives best performance for special purpose applications. Despite all of its advantages it cannot give expected result in general case, so when the applications are of unequal processing time. Another example is Central Manager Algorithm (Grousa and Anthony, 2005).

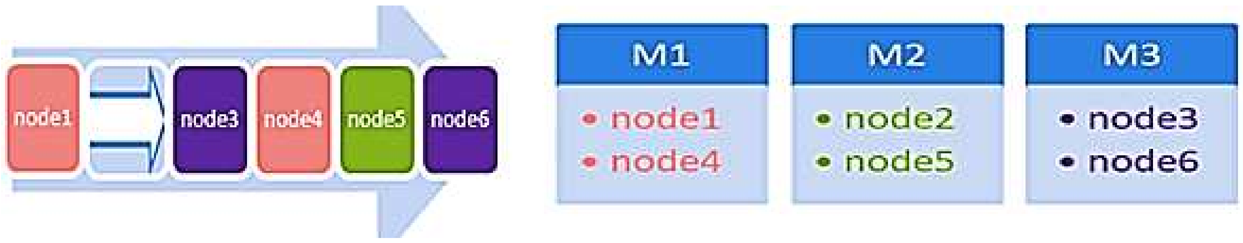

Fig. 5: System with load balancing

In this algorithm a central node selects the slave node for transferring the load. Slave node which has the least load is selected and assigned the application. The central node maintains the load index of all slave nodes connected to it. Whenever, load is changed, a message was send by the slave nodes to the central node. This algorithm needs a high level of inter process communication, which can sometimes lead to the bottleneck state. This algorithm performs better when dynamic activities are created by different hosts.

Discussion

Load balancing is an important task in cloud computing to achieve maximum utilization of system resources. Our article discussed various load balancing methods: Run time and compile methods. On one hand compile load balancing method provides easiest implementation and monitoring of environment but fail to deal with heterogeneous nature of cloud. On the other hand, run time load balancing method is difficult to simulate but is best suited in heterogeneous environment of cloud computing. With run time load balancing, the overhead for storing the previous state of the system is eliminated, because there is no need to have the prior knowledge of the state of the system.

Conclusion

In this study, we illustrate the scheduling system in cloud computing with its different strategies. Each strategy has an important role in improving the cloud system. Researchers should take into account the all parts of scheduling system, because cloud environment needs to be improved from the point of views of all stakeholders. We discuss the types of scheduling methods, fault tolerance methods and load balance methods. The paper also shows algorithms for each type. Because the good fault tolerance algorithm aims to achieve low time and applying load balancing algorithm makes the system more efficient, the scheduler must include these algorithms.

Acknowledgement

Thanks for Prof. Ayman El Sayed, Prof. Gamal Attiya, and Dr. Nermin El-Bahnasawy for their helping.

Author's Contributions

Aida Abou-Elseoud Nasr: Wrote all the research work and contributed to the writing of the manuscript.

Sheren Ahmed: Organized the research work and check Language of the manuscript.

Ethics

There are many of interests in cloud computing resources migration work at the future.

References

Abubakar, H.R. and A. Usman, 2004. Evaluation of load balancing strategie. National Conf. Emerg. Technol.

Aida, A., A. Nirmeen and E.S. Ayman, 2015a. Performance enhancement of scheduling algorithm in heterogeneous distributed computing systems. Int. J. Adv. Comput. Sci. Applic., 6: 88-97.

Aida, N.A.A., N.A. El-Bahnasawy and A. El-Sayed, 2015b. Task scheduling algorithm for high performance heterogeneous distributed computing systems. Int. J. Comput. Applic.

Aida, N.A., N.A. El-Bahnasawy and A. El-Sayed, 2014. Task scheduling optimization in heterogeneous distributed systems. Int. J. Comput. Applic., 107: 5-12. DOI: 10.5120/18737-9982

Aida, N.A., N.A. EL-Bahnasawy and E.S. Ayman, 2016. A new duplication task scheduling algorithm in heterogeneous distributed computing systems. Bull. Electrical Eng. Informat., 5: 373-382.

Bala, A. and I. Chana, 2012. Fault tolerance-challenges, tehniques and implementation in clod computing. Int. J. Comput. Sci.

Bossche, R.V.D., K. Vanmechelen and J. Broeckhove, 2013. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. Future Generation Comput. Syst., 29: 973-985. DOI: 10.1016/j.future.2012.12.012

Chandrasekaran, K., 2014. Essentials of Cloud Computing. 1st Edn., CRC Press, Boca Raton, ISBN-10: 1482205440, pp: 407.

Chaczko, Z., V. Mahadevan, S. Aslanzadeh and C. Mcdermid, 2011. Availability and load balancing in cloud computing. Proceedings of the International Conference on Computer and Software Modeling, (CSM' 11), Singapore, pp: 134-140.

Dong, B., X. Li, Q. Wu, L. Xiao and L. Ruan, 2012. A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers. J. Parallel Distribut. Comput., 72: 1254-1268. DOI: 10.1016/j.jpdc.2012.05.006

Dorigo, M., C. Blum and M. Birattari, 2004. Ant colony optimization and swarm intelligence. Proceedings of the 6th International Conference, Sept. 22-24, Brussels, Belgium.

George, D.I.A and T.L.A. Beena, 2015. Customer Facilitated Cost-based Scheduling (CFCSC) in cloud. Proc. Comput. Sci., 46: 660-667. DOI: 10.1016/j.procs.2015.02.119

Grousa, D. and T. Anthony, 2005. Noncooperative load balancing in distributed systems. J. Parallel Distribut. Comput., 65: 1022-1034.\ DOI: 10.1016/j.jpdc.2005.05.001

Haryani, N. and D. Jagli, 2014. Dynamic method for load balancing in cloud computing. IOSR J. Comput. Eng., 16: 23-28.

Haupt, R.L. and S.E. Haupt, 2004. Parallel Genetic Algorithms. John Wiley and Sons.

Jhawar, R., V. Piuri and M. Santambrogio, 2013. Fault tolerance management in cloud computing: A system-level perspective. IEEE Syst. J., 7: 288-298. DOI: 10.1109/JSYST.2012.2221934

Kaur, J. and S. Kinger, 2014. Analysis of different techniques used for fault tolerance. Int. J. Comput. Sci. Inform. Technol.

Limam, S. and G. Belalem, 2014. A migration approach for fault tolerance in cloud computing. Int. J. Grid High Performance Comput., 6: 24-37. DOI: 10.4018/ijghpc.2014040102

Meshram, A.D., A.S. Sambara and S.D. Zade, 2013. Fault tolerance model for reliable cloud computing. Int. J. Recent Innovat. Comput. Commun., 1: 600-603.

Nazir, B., K. Qureshi and P. Manuel, 2009. Adaptive checkpointing strategy to tolerate faults in economy based grid. J. Supercomput., 50: 1-18. DOI: 10.1007/s11227-008-0245-6

Omara, F.A. and M.M. Arafa, 2009. Genetic algorithm for task scheduling problem. J. Parallel Distrib. Comput.

Shao, Y., 2015. Virtual resource allocation based on improved particle swarm optimization in cloud computing environment. Int. J. Grid Distribut. Comput., 8: 111-118.

Sharma, S., S. Singh and Meenakshi, 2008. Performance analysis of load balancing algorithms. World Acad. Sci.

Venter, G. and G. Venter, 2002. Particle swarm optimization. AIAA J.

Verma, A. and S. Kaushal, 2015. Cost-time efficient scheduling plan for executing workflows in the cloud. J. Grid Comput., 13: 495-506. DOI: 10.1007/s10723-015-9344-9

Wenhong, T. and Y. Zhao, 2015. Optimized Cloud Resource Management and Scheduling: Theory and Practice. 1st Edn., Morgan Kaufmann, Waltham, ISBN-10: 0128014768, pp: 284.

Xu, J., A. Lam and V.O.K. Li, 2010. Chemical reaction optimization for the grid scheduling problem. Proceedings of IEEE International Conference on Communications, May 23-27, IEEE Xplore Press, Cape Town. DOI: DOI: 10.1109/ICC.2010.5502406

Yuming, X.U., K. Li, L. He and T.K. Truong, 2013. A DAG scheduling scheme on Heterogeneous computing systems using double molecular structure-based chemical reaction optimization. J. Parallel Distrib. Comput., 73: 1306-1322. DOI: 10.1016/j.jpdc.2013.05.005