Journal of Engineering and Applied Sciences 13 (11): 3935-3941, 2018
ISSN: 1816-949X
© Medwell Journals, 2018

Load Balancing Algorithm a Comparative Study in Cloud Environment Computing

[1]Manjusha Kalekuri and [2]Kolasani Ramchand H. Rao
[1]Department of Computer Science, Acharya Nagarjuna University, 522006 Guntur, India
[2]Department of Computer Science, ASN Degree College Tenali, Tenali, India

Abstract: Cloud computing is a new trend emerging in IT environment with huge requirements of infrastructure and resources. Load balancing is an important aspect of cloud computing environment. Efficient load balancing scheme ensures efficient resource utilization by provisioning of resources to cloud user's on-demand basis in pay-as-you-say-manner. Load balancing may even support prioritizing users by applying appropriate scheduling criteria. This study presents various load balancing schemes in different cloud environment based on requirements specified in Service Level Agreement (SLA).

Key words: Cloud computing, load balancing, resource provisioning, resource scheduling, Service Level Agreement (SLA), scheduling criteria

INTRODUCTION

Cloud computing is made up by aggregating two terms in the field of technology. First term is cloud and the second term is computing. Cloud is a pool of heterogeneous resources. It is a mesh of huge infrastructure and has no relevance with its name "cloud". Infrastructure refers to both the applications delivered to end users as services over the internet and the hardware and system software in datacenters that is responsible for providing those services. In order to make efficient use of these resources and ensure their availability to the end users "computing" is done based on certain criteria specified in SLA. Infrastructure in the cloud is made available to the user's on-demand basis in pay-as-you-say-manner. Computation in cloud is done with the aim to achieve maximum resource utilization with higher availability at minimized cost.

Cloud v/s cluster and grid: Clusters (Foster *et al*., 2008) are parallel and distributed systems, governed under the supervision of single administrative domain. The node (stand-alone computers) in the cluster integrates to form a single computing resource.

Grid (Foster *et al*., 2008) is aggregation of autonomous resources that are geographically distributed.The nodes in grid permit sharing and selection dynamically at runtime. Clouds (Foster *et al*., 2008; Youseff *et al*., 2008) are not the combination of clusters and grid but are next generation to clusters and grid. Similar to cluster and grid, cloud is also a collection of parallel and distributed systems. Cloud is not a single domain. Unlike cluster and grid, cloud has multiple domains and the nodes of cloud are "Virtualized".

Cloud perspectives: Cloud has different meaning to different stakeholders. There are three main stakeholders of cloud:

End users: These are the customers or consumers of cloud. They use the various services (infrastructure/software/platform) provided by the cloud. Before using the cloud services, the users of cloud must agree to the Service Level Agreement (SLA) specified by the cloud provider. They use the services on demand basis and have to pay for the services availed depending upon their usage. Cloud provides its users flexibility in availing its services by incorporating utility computing. Prior to signing of SLA, the users of cloud must verify that SLA contains certain Quality-of-Service (QoS) parameters which are pre-requisites of the consumer, before using cloud services. Some of the basic requirements or issues of cloud users is listed in Table 1.

Hence, for end user, cloud computing is a scenario where the user can have access to any kind of infrastructure, software or platform in a secure manner-at reduced cost-on demand basis-in an easy to use manner.

Cloud provider: Cloud provider can offer either public or private or hybrid cloud. They are responsible for building

Corresponding Author: Manjusha Kalekuri, Department of Computer Science, Acharya Nagarjuna University, 522006 Guntur, India

Table 1: Stakeholders of cloud

Type of stakeholder	Requirements/Issues
End user	Security
	Provenance
	Privacy
	High availability
	Reduced cost
	Ease-of-use
Cloud provider	Managing resources
	Outsourcing
	Resource utilization
	Energy efficiency
	Metering
	Providing resources
	Cost efficiency
	Meet end user requirements
	Utility computing
Cloud developer	Elasticity/scalability
	Virtualization
	Agility and adaptability
	Availability
	Data management
	Reliability
	Programmability

of the cloud. Private clouds (Zhang *et al*., 2010) are owned by enterprises or business for their internal use. They may use it to store and manage big-data of their organization or to provide enough resources on demand basis to its team of employees or clients. They offer greatest level of security. OpenStack, VMware (Anonymous, 2013) and CloudStack are private clouds.

Public clouds (Zhang *et al*., 2010) may be used by individuals or an organization based upon their requirements and necessities. They offer greatest level of efficiency in shared resources. Confidentiality is the major security issue in using public cloud. They are more vulnerable than private clouds. Amazon web services (Anonymous, 2018), Google compute engine, Microsoft Azure, HP cloud (Anonymous, 2000) are some of the public clouds.

A hybrid cloud (Zhang *et al*., 2010) is a combination of public and private cloud. It allows businesses to manage some resources internally within organization and some externally. The downside is that the complexity of overall management increases along with security concerns. To optimize the use of one or more combination of private or public clouds CliQr (Anonymou, 2000) allows the businesses to accommodate changing needs of users.

Cloud provider must accomplish its job of "resource provisioning". Resource provisioning includes two main tasks. These include managing of huge bundle of resources that make up cloud and providing these resources to the end users. Several provisioning related issues are mentioned in Table 1.

Cloud developer: This entity lies between end user and cloud provider. Cloud developer has the responsibility of taking into consideration both the perspectives of the cloud (i.e., view of end user and cloud provider). The developer of cloud must adhere to all the technical details of the cloud which are essential to meet the requirements of both, the cloud user as well as the cloud provider. Some of the basic issues that cloud developer must focus on are given in Table 1. Main motive of the developer is to bridge the gap between the end user of the cloud and the cloud provider.

MATERIALS AND METHODS

Load balancing in cloud

Computing environment: Load balancing in cloud computing provides an efficient solution to various issues residing in cloud computing environment set-up and usage. Load balancing must take into account two major tasks, one is the resource provisioning or resource allocation and other is task scheduling in distributed environment. Efficient provisioning of resources and scheduling of resources as well as tasks will ensure:

- Resources are easily available on demand
- Resources are efficiently utilized under condition of high/low load
- Energy is saved in case of low load (i.e., when usage of cloud resources is below certain threshold)
- Cost of using resources is reduced

For measuring the efficiency and effectiveness of load balancing algorithms simulation environment are required. CloudSim (Calheiros *et al*., 2011) is the most efficient tool that can be used for modeling of cloud. During the lifecycle of a cloud, CloudSim allows VMs to be managed by hosts which in turn are managed by datacenters.

Cloudsim provides architecture with four basic entities. These entities allow user to set-up a basic cloud computing environment and measure the effectiveness of load balancing algorithms. A typical cloud modeled using CloudSim consists of following four entities datacenters, hosts, virtual machines and application as well as system software. Datacenters entity has the responsibility of providing infrastructure level services to the cloud users. They act as a home to several host entities or several instances host's entities aggregate to form a single datacenter entity. Hosts in cloud are physical servers that have pre-configured processing capabilities. Host is responsible for providing software level service to the cloud users. Hosts have their own storage and memory.

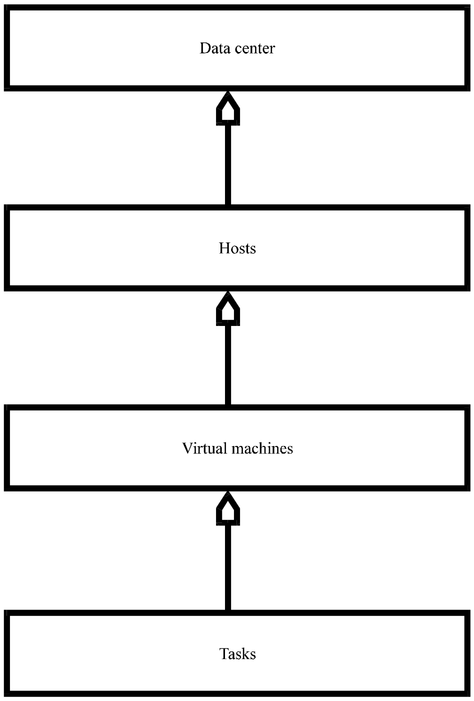

Fig. 1: Class diagram of cloud

Processing capabilities of hosts is expressed in MIPS (Million Instructions Per Second). They act as a home to virtual machines or several instances of virtual machine entity aggregate to form a host entity. Virtual machine allows development as well as deployment of custom application service models. They are mapped to a host that matches their critical characteristics like storage, processing, memory, software and availability requirements. Thus, similar instances of virtual machine are mapped to same instance of a host based upon its availability. Application and system software are executed on virtual machine on-demand.

Class diagram of cloud architecture illustrating relationship between the four basic entities is shown in Fig. 1. Thus, the object oriented approach of CloudSim can be used to simulate cloud computing environment.

Resource allocation: Resource provisioning is the task of mapping of the resources to different entities of cloud on demand basis. Resources must be allocated in such a manner that no node in the cloud is overloaded and all the available resources in the cloud do not undergo any kind of wastage (wastage of bandwidth or processing core or memory, etc.). Mapping of resources to cloud entities is done at two levels:

VM mapping onto the host: Virtual machines reside on the host (physical servers). More than one instance of VM can be mapped onto a single host subject to its availability and capabilities.

Host is responsible for assigning processing cores to VM. Provisioning policy define the basis of allocating processing cores to VM on demand. Allocation policy or algorithm must ensure that critical characteristics of host and VM do not mismatch.

Application or task mapping onto VM: Applications or tasks are actually executed on VM. Each application requires certain amount of processing power for their completion. VM must provide required processing power to the tasks mapped onto it. Tasks must be mapped onto appropriate VM based upon its configuration and availability.

Task scheduling: Task scheduling is done after the resources are allocated to all cloud entities. Scheduling defines the manner in which different entities are provisioned. Resource provisioning defines which resource will be available to meet user requirements whereas task scheduling defines the manner in which the allocated resource is available to the end user (i.e., whether the resource is fully available until task completion or is available on sharing basis). Task scheduling provides “multiprogramming capabilities” in cloud computing environment. Task scheduling can be done in two modes:

- Space shared
- Time shared

Both hosts and VM can be provisioned to users either in space shared mode or time shared mode. In space sharing mode resources are allocated until task does not undergo complete execution (i.e., resources are not preempted) whereas in time sharing mode resources are continuously preempted till task undergoes completion.

Table 2 gives the comparison of resource allocation and task scheduling and specifies the issues resolved by each technique of load balancing. Based on resource provisioning and scheduling four cases can be examined under different performance criteria, so, as to get efficient load balancing scheme.

- Case 1: hosts and VMs both are provisioned in space sharing manner
- Case 2: hosts and VMs both are provisioned to VMs and tasks, respectively in time sharing manner

Table 2: Comparison between resource allocation and task scheduling

Taskk	Sub-category	Issues resolved	Provider oriented	Customer oriented
Resource allocation	At host level At VM level	Efficient utilization Minimize makespan Ensure availability	Yes	Yes
Task scheduling	Space-sharing time-sharing	Minimize overall response time	No	Yes

- Case 3: hosts are provisioned to VMs in space sharing manner and VMs are provisioned to tasks in time sharing manner
- Case 4: hosts are provisioned to VMs in time sharing manner and VMs are provisioned to tasks in space sharing manner

RESULTS AND DISCUSSION

Related work to load
Balancing algorithms: Cloud is made up of massive resources. Management of these resources requires efficient planning and proper layout. While designing an algorithm for resource provisioning on cloud the developer must take into consideration different cloud scenarios and must be aware of the issues that are to be resolved by the proposed algorithm. Therefore, resource provisioning algorithm can be categorized into different classes based upon the environment, purpose and technique of proposed solution.

Load balancing on the basis of cloud environment: Cloud computing can have either static or dynamic environment based upon how developer configures the cloud demanded by the cloud provider.

Static environment: In static environment the cloud provider installs homogeneous resources. Also, the resources in the cloud are not flexible when environment is made static. In this scenario, the cloud requires prior knowledge of nodes capacity, processing power, memory, performance and statistics of user requirements. These user requirements are not subjected to any change at run-time. Algorithms proposed to achieve load balancing in static environment cannot adapt to the run time changes in load. Although static environment is easier to simulate but is not well suited for heterogeneous cloud environment.

Round-Robin algorithm (Sotomayor *et al*., 2009) provides load balancing in static environment. In this the resources are provisioned to the task on First-Cum-First-Serve (FCFS, i.e., the task that entered first will be first allocated the resource) basis and scheduled in time sharing manner. The resource which is least loaded (the node with least number of connections) is allocated to the task. Eucalyptus uses greedy (first-fit) with round-robin for VM mapping.

Radojevic and Zagar (2011) proposed an improved algorithm over round robin called CLBDM (Central Load Balancing Decision Model. It uses the basis of round robin but it also measures the duration of connection between client and server by calculating overall execution time of task on given cloud resource.

Dynamic environment: In dynamic environment the cloud provider installs heterogeneous resources. The resources are flexible in dynamic environment. In this scenario cloud cannot rely on the prior knowledge whereas it takes into account run-time statistics. The requirements of the users are granted flexibility (i.e., they may change at run-time). Algorithm proposed to achieve load balancing in dynamic environment can easily adapt to run time changes in load.

Dynamic environment is difficult to be simulated but is highly adaptable with cloud computing environment. Based on WLC (Lee and Jeng, 2011) (Weighted Least Connection) algorithm, Ren proposed a load balancing technique in dynamic environment called ESWLC. It allocates the resource with least weight to a task and takes into account node capabilities. Based on the weight and capabilities of the node, task is assigned to a node. LBMM (Load Balancing Min-Min) algorithm proposed in study (Wang *et al*., 2010) uses three level frameworks for resource allocation in dynamic environment. It uses OLB (Opportunistic Load Balancing) algorithm as its basis. Since, cloud is massively scalable and autonomous, dynamic scheduling is better choice over static scheduling.

Load balancing based on spatial distribution of nodes: Nodes in the cloud are highly distributed. Hence, the node that makes the provisioning decision also governs the category of algorithm to be used. There can be three types of algorithms that specify which node is responsible for balancing of load in cloud computing environment.

Centralized load balancing: In centralized load balancing technique all the allocation and scheduling decision are made by a single node. This node is responsible for storing knowledge base of entire cloud network and can apply static or dynamic approach for load balancing. This technique reduces the time required to analyze different cloud resources but creates a great overhead on the centralized node. Also, the network is no longer fault tolerant in this scenario as failure intensity of the overloaded centralized node is high and recovery might not be easy in case of node failure.

Table 3: Comparison table of load balancing algorithm in cloud computing environment

Algorithm	Static environment	Dynamic environment	Centralized balancing	Distributed balancing	Hierarchical balancing
Round-robin	Yes	No	Yes	No	No
CLBDM	Yes	No	Yes	No	No
Ant colony	No	Yes	No	Yes	No
Map reduce	Yes	No	No	Yes	Yes
Particle swarm optimiza-tion	No	Yes	No	Yes	No
MaxMin	Yes	No	Yes	No	No
MinMin	Yes	No	Yes	No	No
Biased random sampling	No	Yes	No	Yes	No
Active clustering	No	Yes	No	Yes	No
LBMM	No	Yes	No	No	Yes
OLB (Al Nuaimi *et al*., 2012)	Yes	No	Yes	No	No
WLC	No	Yes	Yes	No	No
ESWLC	No	Yes	Yes	No	No
Genetic algorithm	No	Yes	Yes	No	No

Distributed load balancing: In distributed load balancing technique, no single node is responsible for making resource provisioning or task scheduling decision. There is no single domain responsible for monitoring the cloud network instead multiple domains monitor the network to make accurate load balancing decision. Every node in the network maintains local knowledge base to ensure efficient distribution of tasks in static environment and re-distribution in dynamic environment.

In distributed scenario, failure intensity of a node is not neglected. Hence, the system is fault tolerant and balanced as well as no single node is overloaded to make load balancing decision.

Comparison of different static and dynamic load balancing algorithms is given in Table 3. It also compares them on the basis of spatial distribution of nodes. A nature inspired solution is presented in study (Randles *et al*., 2008) called honey bee foraging for load balancing in distributed scenario. In honey bee foraging the movement of ant in search of food forms the basis of distributed load balancing in cloud computing environment. This is a self organizing algorithm and uses queue data structure for its implementation. Biased random sampling (Randles *et al*., 2010) is another distributed load balancing technique which uses virtual graph as the knowledge base.

Hierarchical load balancing: Hierarchical load balancing involves different levels of the cloud in load balancing decision. Such load balancing techniques mostly operate in master slave mode. These can be modeled using tree data structure wherein every node in the tree is balanced under the supervision of its parent node. Master or manager can use light weight agent process to get statistics of slave nodes or child nodes. Based upon the information gathered by the parent node provisioning or scheduling decision is made.

Three-phase hierarchical scheduling proposed in study (Wang *et al*., 2011) has multiple phases of scheduling. Request monitor acts as a head of the network and is responsible for monitoring service manager which in turn monitor service nodes. First phase uses BTO (Best Task Order) scheduling, second phase uses EOLB (Enhanced Opportunistic Load Balancing) scheduling and third phase uses EMM (Enhanced Min-Min) scheduling.

Load balancing based on task dependencies: Dependent tasks are those whose execution is dependent on one or more sub-tasks. They can be executed only after completion of the sub-tasks on which it is dependent. Therefore, scheduling of such task prior to execution of sub-tasks is in-efficient. Task dependency is modeled using workflow based algorithms.

Workflow basically uses DAG (Wu *et al*., 2013) as knowledge base to represent task dependency. Different workflow based solution consider different parameters. Algorithm are designed keeping in mind whether single or multiple workflows are to be modeled or single or multiple QoS parameters are to be maintained in the system. Different workflows with or without completely different structure are termed as multiple workflows. Workflows can also be classified as transaction incentive (multiple instances of one workflow that have same structure) and data incentive workflows (size and quantity of data is large).

Cost based scheduling algorithm by Xu *et al*. (2009) is designed for single workflows. It partitions the workflows and assigns each partition a deadline. Zhifeng Yu and Weisong Shi designed an algorithm for multiple workflows which focus only on execution time. With an aim of maximizing throughput (Li *et al*., 2011) proposed scheduling strategy which is meant for transaction incentive workflows.

Table 4: Comparison of different types of load balancing scenarios in cloud computing environment

Type of algorithm	Knowledge base	Issues to be addressed	Usage	Drawbacks
Static	Prior knowledge base is required about each node statistics and user requirements	Response time Resource utilization Scalability Power consumption and Energy utilization Makespan Throughput/performance	Used in homogeneous environment	Not flexible Not scalable Is not compatible with changing user requirements as well as load
Dynamic	Run time statistics of each node are monitored to adapt to changing load require-ments	Location of processor to which load is transferred by an overloaded processor Transfer of task to a remote machine Information gathering Load estimation Limiting the number of migrations Throughput	Used in heterogeneous environment	Complex Time consuming
Centralized	Single node or server is responsible for maintaining the statistics of entire network and updating it from time to time	Threshold policies Throughput Failure intensity Communication between central server and processors in network Associated overhead	Useful in small networks with low load	Not fault tolerant Overloaded central decision making node
Distributed	All the processors in the network responsible for load balancing store their own local database (e.g., MIB) to make efficient balancing decisions	Selection of processor that take part in load balancing Migration time Interprocessor communication Information exchange criteria Throughput Fault tolerance	Useful in large and heterogeneous environment	Algorithm complexity Communication overhead
Hierarchical	Nodes at different levels of hierarchy communicate with the nodes below them to get information about the network performance	Threshold policies Information exchange criteria Selection of nodes at different levels of network Failure intensity Performance Migration time	Useful in medium or large size network with heterogeneous environment	Less fault tolerant Complex
Workflow de-pendent	DAG is used to model dependencies of task and can be used to make scheduling decision	Type of workflow Single workflow Multiple workflow Transaction incentive workflows Data incentive workflows Fault tolerance Execution time Makespan Migration time	Used in modeling of task dependencies in any kind of environment (either homogeneous or heterogeneous)	Difficult to model Maintenance of knowledge base is complex Higher complexity

For clouds based on Hadoop CloudWF (computational workflow system) encodes workflow blocks and block-to-block dependencies. Hadoop HBase sparse table is used to store information related to workflows. It is fault tolerant and uses map-reduce framework.

Table 4 compares different type of load balancing scenarios in cloud computing environment. It specifies the knowledge base, usage and drawbacks of each type of algorithm and issues addressed by these algorithms.

CONCLUSION

Load balancing is an essential task in cloud computing environment to achieve maximum utilization of resources. In this study, we discussed various load balancing schemes, each having some pros and cons. On one hand static load balancing scheme provide easiest simulation and monitoring of environment but fail to model heterogeneous nature of cloud. On the other hand, dynamic load balancing algorithm are difficult to simulate but are best suited in heterogeneous environment of cloud computing. Also, the level at node which implements this static and dynamic algorithm plays a vital role in deciding the effectiveness of algorithm. Unlike centralized algorithm, distributed nature of algorithm provides better fault tolerance but requires higher degree of replication and on the other hand, hierarchical algorithm divide the load at different levels of hierarchy with upper level nodes requesting for services of lower level nodes in balanced manner. Hence, dynamic load balancing techniques in distributed or hierarchical environment provide better performance. However, performance of the cloud computing environment can be further maximized if dependencies between tasks are modeled using workflows.

REFERENCES

Al Nuaimi, K., N. Mohamed, M. Al Nuaimi and J.Al-Jaroodi, 2012. A survey of load balancing in cloud computing: Challenges and algorithms. Proceedings of the 2012 2nd International Symposium on Network Cloud Computing and Applications (NCCA), December 3-4, 2012, IEEE., London, UK., pp: 137-142.

Anonymous, 2000. Cisco cloudcenter (formerly CliQr). Cisco Systems, San Jose, California, USA.

Anonymous, 2013. Measuring the business value of VMware horizon view. VMware Computer software company, Palo Alto, California, USA.

Anonymous, 2018. Amazon elastic compute cloud (Amazon EC2). Amazon Web Services, Inc., Seattle, Washington, USA. http://aws.amazon.com/ec2/.

Calheiros, R.N., R. Ranjan, A. Beloglazov, C.A. De Roseand R. Buyya, 2011. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software Pract. Experience, 41: 23-50.

Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud computing and grid computing 360-degree compared. Proceedings of the Workshop on Grid Computing Environments GCE'08, November 12-16, 2008, IEEE., Austin, Texas, USA., pp: 1-10.

Lee, R. and B. Jeng, 2011. Load-balancing tactics in cloud. Proceedings of the 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), October 10-12, 2011, IEEE., Beijing, China, pp: 447-454.

Li, K., G.Xu, G. Zhao, Y. Dong and D.Wang, 2011. Cloud task scheduling based on load balancing ant colony optimization. Proceedings of the 2011 6th Annual International Conference on China Grid, August 22-23, 2011, IEEE., Dalian, Liaoning China, pp: 3-9.

Radojevic, B. and M. Zagar, 2011. Analysis of issues with load balancing algorithms in hosted (cloud) environments. Proceedings of the 34th International Conference on Convention MIPRO, May 23-27, 2011, IEEE., Opatija, Croatia, pp: 416-420.

Randles, M., A. Taleb-Bendiaband and D. Lamb, 2008. Cross layer dynamics in self-organising service oriented architectures. Proceedings of the 2008International Workshop on Self-Organizing Systems Lecture Notes in Computer Science, Vol. 5343, December 10-12, 2008, Springer, Berlin, Germany, pp: 293-298.

Randles, M., D. Lamb and B.A. Taleb, 2010. A comparative study into distributed load balancing algorithms for cloud computing. Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, April 20-23, 2010, IEEE., Perth, Australia, pp: 551-556.

Sotomayor, B., R.S. Montero, I.M. Llorenteand and I. Foster, 2009. Virtual infrastructure management in private and hybrid clouds. IEEE. Internet Comput., 13: 14-22.

Wang, S.C., K.Q. Yan, S.S. Wangand and C.W. Chen, 2011. A three-phases scheduling in a hierarchical cloud computing network. Proceedings of the 2011 3rd International Conference on Communications and Mobile Computing, April 18-20, 2011, IEEE., Qingdao, China, pp: 114-117.

Wang, S.C., K.Q. Yan, W.P. Liao and S.S.Wang, 2010. Towards a load balancing in a three-level cloud computing network. Proceedings of the 2010 3rd IEEE International Conference on Computer Science and Information Technology, Vol. 1, July 9-11, 2010, IEEE., Chengdu, China, pp: 108-113.

Wu, Z., X. Liu, Z. Ni, D. Yuan and Y. Yang, 2013. A market-oriented hierarchical scheduling strategy in cloud workflow systems. J. Supercomputing, 63: 256-293.

Xu, M., L. Cui, H. Wang and Y. Bi, 2009. A multiple QoS constrained scheduling strategy of multiple workflows for cloud computing. Proceedings of the 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, August 10-12, 2009, IEEE., Chengdu, China, pp: 629-634.

Youseff, L., M. Butricoand and D. Da Silva, 2008. Toward a unified ontology of cloud computing. Proceedings of the 2008 International Workshop on Grid Computing Environments, November 12-16, 2008, IEEE., Austin, Texas, USA., pp: 1-10.

Zhang, Q., L. Cheng and R. Boutaba, 2010. Cloud computing: State-of-the-art and research challenges. J. Internet Serv. Applic., 1: 7-18.