http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (12): 2419-2425, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.2419.2425

Research on Resource Scheduling Algorithm Based on ForCES Network

[1,2]Dai You, [1]Bin Zhuge, [1]Guowei Dai, [2]Weiming Wang and [2]Julong Lan
[1]College of Information and Electronic Engineering, Zhejiang GongShang University,
[2]Institute of Information Engineering, Information Engineering University,
Zhengzhou, 450002, People's Republic of China

Abstract: In view of the problem about how to schedule resource in ForCES network reasonably, maps the resource of ForCES network to CE (Controller Element) through the ForCES protocol by using virtualization technology. The detailed step is that every Agent LFB (Logic Function Block) in FE promptly reports their resource utilization to CE. Accordingly, CE can create a table for update the utilization of each FE (Forwarding Element), to schedule request resource for service conveniently. Through the process of scheduling, two basic factors that need to consider are the completion rate before deadline and cost control. Therefore, the study proposes an algorithm: considering the real-time resources utilization of AgentLFB in FE, CE will schedule resource based on the two aspects of deadline and cost. Simulation results show that this algorithm is not only beneficial to the resource load balancing of ForCES network but also completes the task within deadline with less cost.

Key words: ForCES network, cloud computing, resource scheduling

INTRODUCTION

The traditional Internet structure follows the design principle-"core is simple and edge is smart", so the network intelligence is always deployed at the terminal of network edge. However, with the rapid development of network technology, a large number of new access technologies come out, as well as new heterogeneous network and new applications. These new technologies and applications not only promote the progress of the entire field of communication but also make the traditional Internet structure face enormous challenges (Akhtar, 2007). In addition, the user who wants to establish a set of IT system has to purchase the infrastructure such as broadband, hardware and the appropriate software. Moreover, user also needs a professional team of workers who can maintenance the IT systems. Therefore, a large number of hardware, software, personnel and maintenance will bring a high expenditure to users.

The emergence of cloud computing not only makes a good integration of existing heterogeneous network resources but also provides a large number of new services easily (Luo *et al.*, 2011; Pandey *et al.*, 2010). So, users do not need to worry about the high costs of hardware and workers for the completion of their business any more. And task scheduling is inevitable in the implementation of cloud computing which affects the utilization of resources and the overall effectiveness directly. In the cloud computing system, cloud users only care about whether their requirements can be guaranteed, whether their interests can be achieved and whether the service is convenient; providers of cloud computing also hope to meet there users and reduce overhead costs to gain more profit at the same time via some certain scheduling policies. At home and abroad, many studies have concentrated resource scheduling on cloud computing through various algorithms to maximize the ability of cloud computing hardware resources.

Though study, ForCES is a new kind of network structure. The idea, separation of control elements and forwarding elements, provides conditions to achieve virtualization. And in ForCES systems, virtual technology application is in favor of a better the fault redundancy and resource management. So the study want to combine the existing ForCES structure with cloud computing. That is, the study makes use of virtualization technology to map and deploy physical resources with rational management to use resources more efficiently. The core of parallel distributed systems based on the ForCES framework is that CE schedules and manages LFB resource of multi-FE through ForCES protocol.

RESOURCES SCHEDULING MODEL BASED ON FORCES FRAMEWORK

ForCES network resource model: Now, the content of Internet becomes richer, not only widely cognitive access, routing and so on but also the computing, storage,

Corresponding Author: Dai You, Zhejiang GongShang University, College of Information and Electronic Engineering, Zhengzhou, 450002, People's Republic of China

service, software and other elements. The content of cloud computing not only contains the network but also those things once depicted outside cloud. As using the cloud to depict network for emphasizing the application of network rather than its implementation details, cloud computing use cloud to describe information service infrastructure (network, computing, storage etc.) and software (operating system, application platform, Web services, etc.). The aim is to emphasis on the application of these resources rather than their implementation details. Therefore, uses FE model to describe the heterogeneous resources of cloud computing. FE model based on ForCES framework can effectively express all kinds of resources of cloud computing and satisfy the dynamic expansion and configuration. The resources inside FE are expressed as different kinds of Logic Function Block (LFB). LFB and its properties can be controlled by CE through ForCES protocol. And each LFB connects mutually, the connection relations is controlled by CE through ForCES protocol to form different LFB topology structure, even to realize dynamic resource allocation which can provide data channel for different treatment of the cloud computing application. To monitor and control a whole FE, FE model unifies and simplifies LFB operation and write the defined LFB into standard document of "LFB library". Therefore, the above LFB model has the ability of effectively describer the resources of cloud computing.

The ForCES virtualization technology can map the ForCES on the physical network resources to the CE side which makes CE has the ability to deploy LFB in FE flexibly. And the study uses the network equipment to complete a variety of different services business by constructing different LFB topology. As is shown in Fig. 1, the study design resource management structure model based on ForCES network which consists of a Control Element (CE), Forwarding Elments (FE) and Logic Function Block (LFB). CE is the management center of entire ForCES network; FE and LFB are the network resources and FE is composed of various LFB to meet different business needs. On the other hand, research on ForCES network resource allocation can make a more efficient and more reasonable use of the existing resources of ForCES network.

AgentLFB which is located in FE, is different from the general LFB resources. It is used to collect the available resources in the FE and to report real-time resource usage of its parent FE-CE via ForCES protocol. CE will generate a table based on these conditions and save the real-time usage of every resource, so that CE can be more convenient and more accurate to decide resource allocation.

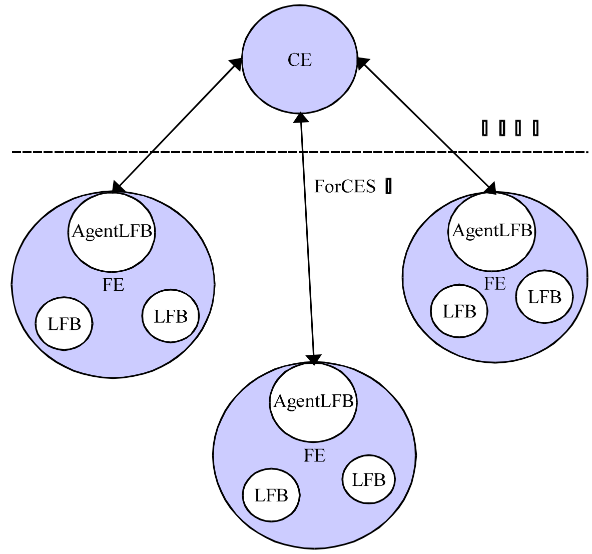

Fig. 1: Structure of the resource model based on ForCES network

FORCES RESOURCES SCHEDULING MODEL BASED ON CLOUD COMPUTING

Resource provider: Cloud computing resources can be regarded as a resource pool. The resource providers are responsible for providing resources, statistical detailed information about these resources send it to the Broker. After Broker receives the call requests, it will ask corresponding resource providers to offer the appropriate computing resources the user from the resource pool.

The first thing that resource provider need to realize is the sharing of all resources on Internet to eliminate resource island. To share these resources, you must first get to know the existence information of the location of resources and their distribution throughout the Internet. The actual existence of a shared resource that can be used to provide services in the network information is contained in the ForCES resource layer. It includes: computing resources (CPU), storage resources (memory, hard disk, etc.), I/O (printers, etc.) and so on. In ForCES network, LFB will have a unified description about resource and report it to CE through AgentLFB. To make an easier analysis, the only factor that considers is the computing resources in scheduling algorithm.

Resource consumers: User can submit a task with described information to the Resource Broker. The description and performance of the task has a direct impact on the user's satisfaction.

The main information of the task that user submitted to the broker can be conclude as the budget on this task,

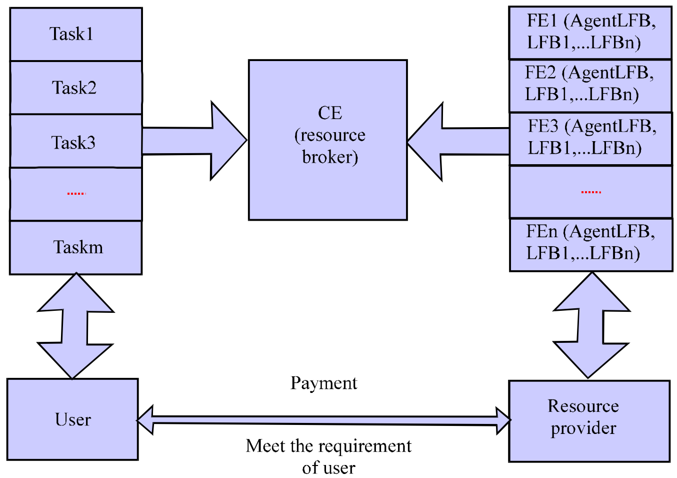

Fig. 2: ForCES resource scheduling model based on cloud computing

deadline of the task and so on. Each job contains the following features:

- Type of task, in favor of task completion time or task completion cost
- Length of task, input and output data of task, the execution start and finish time of the task as well as the owner of the task
- The deadline and budget of task

Resource broker: The main information of each resource that user provided to the broker are the host IP address, the computing capacity of the resources, initial bids (the processor executes prices per second), resources allocation strategies (time-sharing or space sharing), node load and so on.

Resource Broker is the core part of the scheduling model. It disposes the current resources available information which is the hub between the user and resource. Resource broker is able to find the real-time available numbers of resources and the request completion tasks to map the resource to tasks.

In the ForCES architecture, CE acts as a resource broker which can have a global management of unify description LFB resource in multi-FE via ForCES protocol, including configuration, dynamic update. From this perspective, ForCES is a distributed system which can have a unified management of resources. And distributed resources can be registered to the system, or deleted as a unit by FE which can also be registered to the system or deleted by single LFB, so that it can achieve resources virtualization, distribution and dynamic scalability of the system.

ForCES resource scheduling model based on cloud computing is shown as Fig. 2. The study only considers node computing resources. Create an AgentLFB for each FE that can be used to share as a computing resource. AgentLFB contains the main information of resources such as the host's IP address, resource computing power, initial bid and the node load and so on. And then, it will report them to CE end for a better scheduling. Broker matches the registered information of resource with tasks that user submitted via the scheduling algorithm.

DESIGN OF RESOURCE SCHEDULING ALGORITHM BASED ON ECONOMICS

In cloud computing, resource utilization and allocation efficiency are very important which directly affect the overall performance and benefits of cloud computing platform. In Cloud computing, resource scheduling system will map the task set of user $T = \{t_1, t_2, \ldots, t_m\}$ to the resource set $R = \{r_1, r_2, \ldots, r_n\}$ for executing in terms of some scheduling strategies based on the state information and forecast of resources and return the running result to user.

The basic starting point of ForCES resource scheduling model based on cloud computing is: users can submit task with additional requirement to scheduling system which will find reasonable resource for matching based on the tasks requirement user submitted and complete the task in a given deadline and cost.

In this study, scheduling algorithm based on economics aims at achieving a service price according to the user (submitting tasks) and resource provider, so that the most preferential price and resource that meet the requirements is finally elected (Chen *et al*., 2006; Liu *et al*., 2008). If the task deadline is more relaxed, the chance that users get the lower price resources will be relatively larger, the cost of using resources will be relatively lower; conversely, if the task deadline is more urgent, users need to buy the higher-priced resources within maximum fee that they can pay to ensure the task can be completed in time (Gong *et al*., 2009).

Next, introduces two classical scheduling algorithms.

Min-Min algorithm: The basic idea of Min-Min algorithm is according to the strategy of minimum and earliest completion time, cloud computing system matching the task t_i in tasks set T received within scheduling time slot with the resource r_j in resources set R (Ma and Liu, 2012). The description of this algorithm is as follows:

Step 1:	If task queue T is not empty, then continue to execute, or execute (13)
Step 2:	For (all tasks t_i in task set T)
Step 3:	For (all resources r_j in resource set R)
Step 4:	$C_{ij} = tr_{ij}+tw_{ij}$
Step 5:	End for
Step 6:	End for
Step 7:	For (every elements in C)
Step 8:	Searching task t_i with minimum and earliest complete time, relative resource r_j
Step 9:	End for
Step 10:	Assigning tasks t_i to resource r_j
Step 11:	Removing task t_i from T
Step 12:	Updating tw, return to (1)
Step 13:	Exiting loop

The tw_{ij} stands for the preparation time that task t_i waiting for resource r_j, task t_i will enter the ready state after preparation time. The tr_{ij} represents the execution time that task t_i running in resource r_j, the preparation time and execution time of task constitute the completion time C_{ij} of task. tr_{ij}. C_{ij} is a matrix for storing completion time that task t_i running in resource r_j. Because the Min-Min algorithm uses the scheduling strategy of minimum and earliest completion time, it assigns the task to appropriate resource for execution through the gradual cyclic manner and every scheduling will select a task with minimum completion time, then assign it to the corresponding resources to perform, so the waiting time of small tasks often is short, small tasks can be assigned a high performance resource. When using this algorithm for scheduling, the high performance resources will have a large number of tasks in the queue, the resource with weak performance are almost in the idle state which will cause the serious unbalance load (Wang *et al*., 2010).

Max-Min algorithm: The basic idea of Max-Min algorithm is according to the strategy of maximum and earliest completion time, cloud computing system matching the task t_i in tasks set T received within scheduling time slot with the resource r_j in resources set R (Doulamis *et al*., 2007). The description of this algorithm is as follows:

Step 1:	If task queue T is not empty, then continue to execute, or execute (13)
Step 2:	For (all tasks t_i in task set T)
Step 3:	For (all resources r_j in resource set R)
Step 4:	$C_{ij} = tr_{ij}+tw_{ij}$
Step 5:	End for
Step 6:	End for
Step 7:	For (every elements in C)
Step 8:	Searching task t_i with the maximum and earliest complete time, relative resource r_j
Step 9:	End for
Step 10:	Assigning tasks t_i to resource r_j
Step 11:	Removing task t_i from T
Step 12:	Updating tw, return to (1)
Step 13:	Exiting loop

Max-Min algorithm is similar with Min-Min algorithm. The difference between them is the Min-Min algorithm finds the tasks with minimum and earliest complete time but Max-Min algorithm finds the tasks with maximum and earliest complete time. Max-Min algorithm assigns resources to the large tasks firstly which solves the problem that Min-Min algorithm may lead to imbalance of assigning resource to a certain extent. Because Max-Min algorithm schedules the task with longer running time to the resources with fastest complete time, it makes the larger task and small task can execute simultaneously. However, it also may lead to small tasks waiting for a long time, even the task cannot be completed within the limited time.

SCHEDULING ALGORITHM BASED ON DEADLINE AND COST

This study proposes a scheduling algorithm named Deadline and Cost Based Algorithm (DCBA). This algorithm is based on the scheduling strategy of deadline and cost. Firstly, sorts each task in task set according to the degree of urgency of tasks. After estimating the resources required by task, should match the computing resources of cloud computing resource pool with each task in task set. The degree of urgency is the difference between deadline of each submitted task and the complete time running various resources. All tasks in task set are sorted based on the degree of urgency and are saved in the task queue T to complete the initialization of task set. Then all resources in resources set are sorted based on the computing ability of resources to complete the initialization of resources set. Then extract the task based on the degree of urgency from the task set, match with the resource in cloud computing resource pool and select the resource that can complete task in the shortest time to ensure the task can be completed as early as possible within deadline. If satisfied resource is more than one, then select the resource with least cost.

In task queue of cloud computing, all users hope their task can be completed within their expected time. However, because of the difference of service level they purchased, task will be different in the ratio of task length and deadline which reflects the feature of task which is scheduled based on the service level in cloud environment. This algorithm introduces the longest waiting time $T_{deadlinei}$ of task to ensure the service quality of the users can be guaranteed. Have $T = \{t_1, t_2, \ldots, t_m\}$ as task set, $R = \{r_1, r_2, \ldots, r_n\}$ as resource set and m represents the numbers of task, n represents Broker gets the numbers of resource from cloud computing resource pool. In Task $t_i = (T_{lengthi}, T_{deadlinei})$, $T_{lengthi}$ represents the size of task, $T_{deadlinei}$ represents the $T_{deadlinei}$ of task t_i. In resource $r_j = (R_{mipsj}, R_{costj})$, R_{mipsj} represents the computing ability of resource r_j, R_{costj} represents the cost per unit time leased resource r_j. When starting for matching resources, tw_j represents the expected time of resources, namely over the period of tw_j, Resource r_j will complete all tasks to enter the ready state. tr_{ij} is also represents an expected time, namely the expected execution time task t_i running in resource r_j. TL is a n×m matrix, it used to record the expected surplus time of each task ti on resource rj, sorted the execution sequences of task based on the surplus time, get task to execute by order. When the task is completed, removing the tasks from the task set T, repeat until the number of task in task set T is 0. Every time to traversal the matrix TL, map task tk with minimal surplus time to a corresponding resource, then update matrix TL and vector tw. If a task set that has two or more tasks conflict for the same surplus time, then the algorithm will firstly allocate the resources to the larger tasks.

The description of algorithm:

Step 1:	Initializing the task set T
Step 2:	Initializing the resources set R
Step 3:	Initializing tr_{ij}, tw_j, c_{ij} and let them equals 0
Step 4:	While (number of task in task set T is not 0) do
Step 5:	For (every task t_i in task set T) do
Step 6:	For (every resource rj in resources set T) do
Step 7:	$C_{ij} = tr_{ij}+tw_j$, Tl_{ij} = Tdeadlinei-C_{ij};
Step 8:	End for
Step 9:	End for
Step 10:	For (every element in TL_{ij})
Step 11:	Searching the minimum task t_k in TL_{ij}£¬for the tasks with the same value, firstly selecting the larger task
Step 12:	End for
Step 13:	For (every resource rj in resource set R) do
Step 14:	Traversing the complete time of task t_k in all resources, assigning the resource r_k with minimum complete time to task
Step 15:	If there are some resources with same complete time, then choose the resources with less costly to task
Step 16:	End for
Step 17:	Searching the task t_i with minimum and earliest, the corresponding resource r_j
Step 18:	Removing t_k from T
Step 19:	Updating tw, return to (4)
Step 20:	End while

EXPERIMENT AND SIMULATION

This study uses cloud computing simulation software CloudSim (Calheiros *et al*., 2009, 2011) as the simulation platform, compare DCBA and Min-Min algorithm, Max-Min algorithm from two aspects to finish simulation: the complete state of task within a given deadline and total cost. There are a group of tasks (Numbers in front of brackets represents the length of task and numbers at the back of brackets represents the deadline of task): t1 (100, 3), t2 (300, 5), t3 (500, 9), t4 (800, 12), t5 (1000, 10), t6 (1500, 12). The computing resources in cloud computing resource pool (Numbers in front of brackets represents the MIPS of resource, numbers at the back of brackets represents the cost for using this resource): r1 (100, 1), r2 (300, 3).

From the Fig. 3, shows complete times of each task based on three algorithms. DCBA algorithm can complete the task well before deadline, Min-Min algorithm also can complete the task before deadline but the completion rate of Max-Min algorithm is relatively low. Because DCBA assigns resources based on the surplus lever of deadline, this ensures that the task will be completed as soon as possible before deadline. However, Min-Min algorithm has the priority to schedule the small task with resource have more powerful, when it processes a group of task including small task with shorter deadline and big task with longer deadline, complete rate of task on time is relatively high. Max-Min algorithm has the priority to schedule the larger task with resource have more powerful, so it processes a group of task including small task with longer deadline and big task with shorter deadline, complete rate of task on time is relatively high. On the other hand, the complete time of DCBA algorithm and Max-Min algorithm is shorter than Min-Min algorithm. This is because when DCBA algorithm and Max-Min algorithm scheduling the larger task to resource

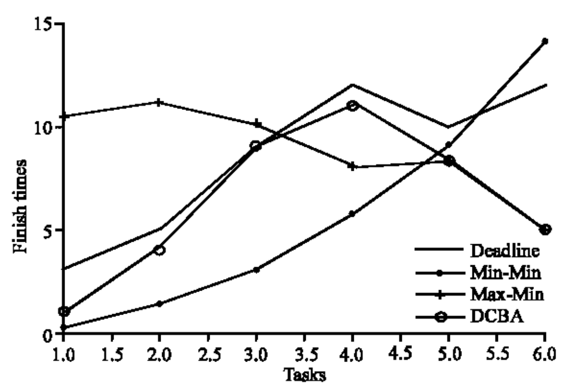

Fig. 3: The deadline of each task and the actual finish time based on each algorithm

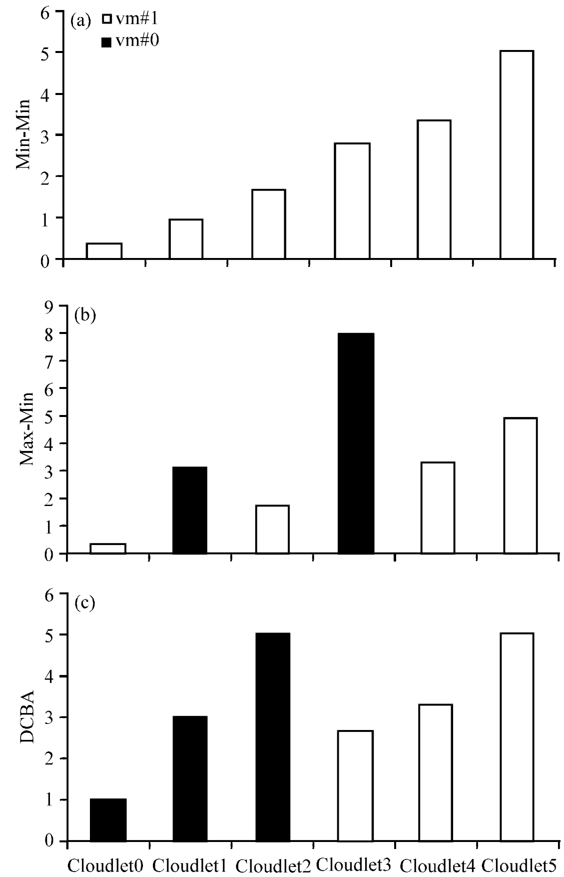

Fig. 4(a-c): The state of each task using resource based on each algorithm (a) Min-Min algorithm, (b) Max-Min algorithm and (c) DCBA algorithm

have more powerful, small tasks would be assigned to the resources with lower processing power. However, Min-Min algorithm assigns small task to the resources with higher processing power, large task will waiting for the resource, so the complete time of entire group will longer.

From Fig. 4a-c shows the state of each task using resource based on three algorithms. In Fig. 4a, Min-Min algorithm assigns all tasks to vm #1, in Fig. 4b Max-Max algorithm and in Fig. 4c, DCBA algorithm is balanced assigns all tasks to vm #0 and vm #1. In terms of costs, the cost of Min-Min algorithm is 14.1*3 = 42.3; the cost of Max-Min algorithm is 11.12+10.43*3 = 42.41; the cost of DCBA algorithm is 9+11*3 = 42. In all, DCBA algorithm also has advantages in cost.

CONCLUSION

This study proposes the resource model based on ForCES network structure. All of the AgentLFB in FE will promptly report their resource utilization to CE and then CE scheduling the resource for service request based on the resource utilization. Then, this study studies DCBA algorithm based on this model. From the simulation, can see that DCBA has advantages in task completion rate within deadline, resource utilization and cost control over Max-Min algorithm and Min-Min algorithm.

ACKNOWLEDGMENTS

This work was supported in part by a grant from the National Basic Research Program of China (973 Program) under Grant No. 2012CB315902, the National Natural Science Foundation of China under Grant No. 61102074, 61170215, the Program for Zhejiang Leading Team of Science and Technology Innovation under Grant No. 2011R50010, the Program for Zhejiang Sci and Tech Project under Grant No. 2012C33076, the Program for Zhejiang Provincial Natural Science Foundation of China under Grant No. Y1111117, Q12F02013.

REFERENCES

Akhtar, Z., 2007. Genetic load and time prediction technique for dynamic load balancing in grid computing. Inform. Technol. J., 6: 978-986.

Calheiros, R.N., R. Ranjan, C.A.F. De Rose and R. Buyya, 2009. CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services. http://arxiv.org/ftp/arxiv/papers/0903/0903.2525.pdf

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

Chen, D.E., Y. Yang and L. Liu, 2006. Utility optimization-based grid computing resource scheduling algorithm. Comput. Eng. Des., 42: 191-193.

Doulamis, N.D., A.D. Doulamis, E.A. Varvarigos and T.A. Varvarigou, 2007. Fair scheduling algorithms in grids. IEEE Trans. Parallel Distrib. Syst., 18: 1630-1648.

Gong, H., J. Yu, Y. Hou and H. Liu, 2009. User QoS and system index guided task scheduling in computing grid. Comput. Eng., 35: 52-54.

Liu, S.Q., H.L. Shao and X.L. Feng, 2008. Resource scheduling strategy based on credit in computational economy model for grid. Comput. Eng. Des., 29: 3836-3838.

Luo, J.Z., J.H. Jin, A.B. Song and F. Dong, 2011. Cloud computing: Architecture and key technologies. J. Commun., 32: 3-21.

Ma, L. and G.Y. Liu, 2012. Improved min-min scheduling algorithm. Comput. Eng. Appl., 48: 69-73.

Pandey, S., L. Wu, S. Guru and R. Buyya, 2010. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, April 20-23, 2010, Perth WA, Australia, pp: 400-407.

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 3rd IEEE International Conference on Computer Science and Information Technology, July 9-11, 2010, Chengdu, China, pp: 108-113.