A Greedy Approach for Optimizing the Problems of Task Scheduling and Allocation of Cloud Resources in Cloud Environment

Asif Mohammad[1]
Prof. Ashish Kumar[2]
Lal Shri Vratt Singh[3]
[1]*Research Scholar, ITS Engineering College, Gr. Noida*
[2] *Professor, ITS Engineering College, Gr. Noida*
[3]*Research Scholar, Jamia Millia Islamia, New Delhi*
[1]*asifkl786@gmail.com*
[2]*ashishkumarcse@its.edu.in*
[3]*lsvslalit@gmail.com*

ABSTRACT: *Cloud computing is a computing technique which is an on demand mechanism having pay per usage, and service based technique where elastic and scalable IT functionalities are delivered to several user ends using world wide web (internet) as a communication medium. Distributed computing strategy, utility based approach and virtualization also play a major role to provide an efficient and efficient base for the creation of the new concept of cloud computing. It provides a critical virtual pool of associated computing resources which are accessed and used by the end users over the medium i.e. internet. After looking the revolutionary benefits and advantages of cloud computing approach, the market is going to shift from previous traditional technologies to new cloud computing based approaches and technologies. Resource allocation is a prim concern which plays an important role in many computational fields, such as in data-center management and in operating systems. In cloud based systems, resource allocation part can be said as a mechanism that looks to ensure that the computing requirements of the applications are properly, completely and correctly fulfilled by the server's infrastructure. The servers which provides the services of cloud computing, keep this most important aspect in their notice that the resources are to be utilized efficiently so that they may generate maximum profit. This outputs the allocation of resources and the scheduling of tasks as a prim and core challenging issue in cloud computing.*

This paper comes with some effective and efficient task scheduling and resource allocation algorithms which output the optimum results while implemented in cloud computing.

KEYWORDS

Cloud Computing, Information Technology, Task Scheduling Algorithm, Optimized Algorithm, Activity Based Costing (ABC), Resource Allocation Algorithm.

1. INTRODUCTION

While talking about on-demand services, the modern approach of computing i.e. cloud computing has been emerged as very much popular and revolutionary approach in IT industry to support and operate dynamic platform of computing. It is a new computing style where elastic, integrated and scalable modules and IT resource functionalities are resulted to the clients using world wide web as a service. In the cloud computing environment, the scheduling strategy confirms the good selection of existing cloud resources for efficient, proper, effective and successful execution of the incoming tasks after considering the applied restrictions and looking static/dynamic nature of subsequent tasks. Over recent years as the activities and involvement of cloud computing in IT sector are becoming very popular, the different-different scheduling mechanisms of cloud computing are receiving a big focusing attention. Basically, the scheduling can be said as a process or mechanism in which the various tasks are mapped one to one and/or by various ways to the vacant resources after finding the requirements and associated characteristics of the tasks. It is very essential aspect while working with cloud involved environment that various parameters associated with tasks need to be kept into notice in order to perform efficient and effective scheduling.

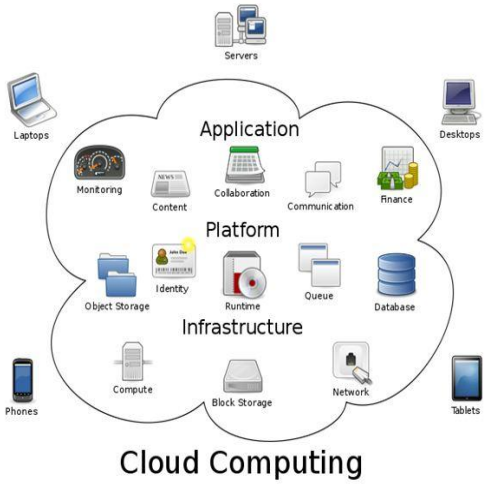

Figure 1: Cloud computing architecture

2. PROBLEMS ASSOCIATED WITH CLOUD RESOURCE ALLOCATION & SCHEDULING OF TASKS

2.1. Cloud Resource Allocation

The cloud resources for computation are required to the client/user for a certain period of time for a task. For e.g., a large number of systems can be required for a researcher to simulate or compute a project for few seconds or minutes or hours, but it might not be essential, at any specific duration of time. A telecom company can process its pre-existing setup which allows the database accessibility and the hosting of various websites, but may be essential to substitute that establishment in addition with some more resources while the web traffic got suddenly increased. In other words, those additional resources have to be kept ready and always available right away after a short advance notice[1]. The different resource allocation mechanisms have been implemented in many areas of modern computing, such as client-server computing model, grid computing, distributed access computing model, operating systems and in the management of datacenters. In the cloud computing environment, the module of resource allocation can be seen as the mechanism which guarantees the good mapping between the providers' and applications' infrastructure and requirements. Apart from this guarantee to the developer, the present status of the each and every resource should be considered and maintained in the cloud environment by the resource allocation strategy so that the algorithm's deployment may allocate the physical and/or virtual resources effectively & efficiently for minimizing the operational cost. [2]

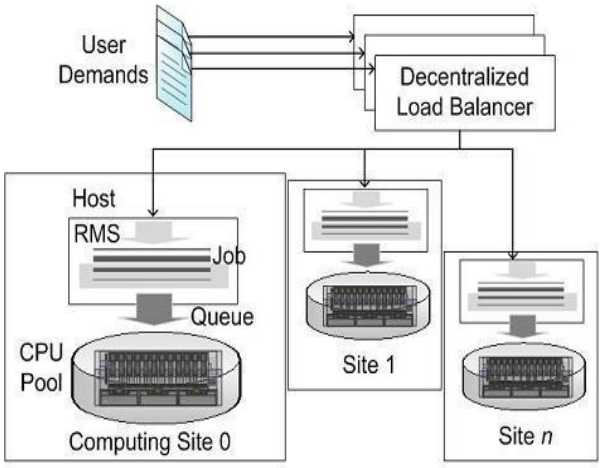

Figure 2: The Structure of the simulated cloud system

Virtual Machine (VM) Allocation:

Allocation of VMs can be separated into halves. The first half separates the new requests for VM provisioning and the VM transferring on hosts. The second half looks on optimizing the current VM allocation. If we try to find the algorithmic complexity of the allocation, it would be in the range of '$n_x m$' which can be determined and analyzed by computing the number of VMs *'n'* which are to be allocated for the future coming tasks and the number of hosts *'m'*. Moreover, the optimization of VMs' current allocation is carried out in two parts- (i) selection of the VMs that need to be allocated for transferring and (ii) placement of the selected VMs on the hosts using allocation algorithm.

2.2 Problems in Scheduling of Tasks

2.2.1 Cloud Service Scheduling

The scheduling of cloud services can be distinguished at two levels, first 'user level' and second 'system level' [3]. If the problems are pointed out between the service providers and customers due to any provisioning related to service, it is said the scheduling at 'user level' while within the datacenter, when the resource management is considered, it is said the scheduling at 'system level'. A datacenter is associated and integrated with various physical machines and a huge number of tasks are received, allocated and assigned to the cloud's physical machine within datacenter. The performance parameters of the datacenter are very closely associated with these assigned allocations. In addition to cloud system utilization, some other quality factors such as the quality of services (QoS), sharing of cloud resources (SCR), real time satisfaction (RTS) and reliability according to service level agreement (SLA) etc., shouldn't be skipped from consideration.

2.2.2 User Level Scheduling

To regulate and monitor the demand and supply of the system's cloud resources, the most suitable and appropriate schedulers are 'auction-based schedulers' and well known 'market-based schedulers'. In the cloud computing scenario, the best effective allocation is carried by the 'market-based resource allocation'. In market-based resource allocation the client gets the virtualized resources (VR) as a service.

A chunk of market-oriented task scheduling algorithms (TSA) to an AuctionNet for spreading network like heterogeneous distributed environment (HDE) is proposed in [4].

Applications of pricing model development (PMD) having dependency consideration in cloud systems approached by two sets of profit-driven scheduling algorithms aimed with composite services using the sharing of processors are proposed in [5].

2.2.3 Static and Dynamic Scheduling

In the static scheduling approach the data elements are pre-fetched, transferred and pipelined to the various stages of the incoming task execution process. This scheduling encounters relatively less overheads and runtime. On other hand, in the

dynamic scheduling approach, the task-components are quite dynamic in occurrence and are not certain before their execution. This is the reason, the processing time of the tasks cannot be pre-determined in advance and the allocation of executable tasks is assumed on fly as the application got executed. An cloud environment of such task executions that avoid the scalable static scheduling techniques (SSST), provides a stretchable pricing model to the client which minimizes the scheduling overheads, are discussed in [3]. In three-tier structured cloud domain, the approach of service request scheduling (SRS) having service and cloud resource providers along with clients should fulfill the needs of end users and service providers as well. A dynamic feedback algorithm is elaborated in [5] for the effective and pre-emptable task scheduling approach. For the task scheduling activity, an efficient optimized task scheduling algorithm (TSA) and its implementation are given in [6] that is based on the activity based costing (ABC). In the hybrid cloud environment (HCE), an experiment is explained in [7] on various strategies of optimized cost-optimal dynamic scheduling (OCDS).

For the cloud systems, an improved and efficient Backfill Algorithm (BA) is proposed in [8] for targeting the quality of services. This algorithm considers the balanced spiral method (BSM) for scheduling. This paper has discussed and analyzed various scheduling algorithms which works to support parallel task scheduling such as EASY and CBA. A quite different scheduling algorithm which considers both aspects- computation performance of cloud systems and cost of the involved resources, is proposed in [9]. This algorithm integrates the user requested tasks according to the power and capability of processing of a dedicated cloud resource and pushes the integrated tasks to the system's cloud resources and also increases the cloud computation to cloud communication (C2C) ratio. Due to the integrating of tasks, the cloud computation to cloud communication (C2C) ratio is getting along optimized by the frequent communication of system's resources and incoming coarse-grained (CG) tasks.

The tremendous and continuous emission of heat, carbon dioxide (CO_2) and the huge energy from the cloud computing servers (CCS) result the enormous thermal pollution in the earth's environment. The green task scheduling (GTS) mechanisms which significantly decrease the pollution by reducing the above stated issues are described in [10] and [3].
The scheduling algorithms which look the issues, concerns and parameters other than stated above are given in [6] and [7]. Over the network, the detailed information of the execution nodes which are vacant and available, is aggregated via the fully decentralized cloud scheduler and is utilized to assign tasks to those available execution nodes which are ready to execute them in time without delay. The study of cloud communication model (CM) and realistic network topology (RNT) propose Deadline Reliability Resource (DDR) aware scheduling algorithm in [5]. An algorithm which emphasizes on reinforcement learning is implemented in [11] that focuses on cloud recovery and failure aspects of the entities in cloud driven systems. It makes and considers the scheduling errors tolerable while the entities are revoked and fetched for a long time. A recent framework of task scheduling technique in tree network is properly documented in [10].

2.2.4 Workflow Scheduling

Within a directed acyclic graph (DAG) the algorithmic structure of the associated applications is enabled over its workflow [3], where its nodes and edges indicate the constituent incoming tasks and their mutual inter-dependencies to the executable applications [4]. In the work flow of scheduling, each and every executable task can be communicated with another executable task and a group of related executable tasks form a single workflow. In the cloud computing systems, a detailed survey on various workflow scheduling algorithms (WSA) is explained in [5]. A detailed study of various task scheduling algorithms (TSAs), issues and problems associated to the cloud computing workflow is elaborated in [11] while [12] tells the nature of instance and its intensive workflows within the cloud systems.

3.RELATED WORK AND SOLUTIONS

3.1 Resource Allocation Algorithm (RAA)

The optimal and efficient use of the cloud resources is carried out very cleverly with the help of the platform-placement algorithms. Eucalyptus which is openly available, uses an algorithm which significantly pays a little attention on the assignment of VMs within the cloud systems. In other hand Open Nebula shows an interesting mechanism for the assignment of VMs considering its rank assessment. Further this mechanism implements many useful policies. Nimbus provides Workspace Pilot (WP) and Workspace Resource Manager (WRM) that are responsible to take the responsibility for the placement of tasks and nodes respectively within the cloud network.

As such, those providers who look the efficient optimization of cloud resource usage as a very much critical requirement, would be more interesting to discuss in [2].

For IaaS clouds, the processes scheduling and resource allocation (SRA) are two key process which consider the VMs as scheduling units, that are assigned to physical resources of the heterogeneous support. To verify, validate and confirm the resource allocation within the cloud systems, Eucalyptus runs Round Robin (RR) and First-fit algorithms (FF) while Nebula executes Preemption Scheduling (PS), Advance Reservation (AR) and Queuing System algorithms (QSA).

Before looking task scheduling process on cloud resources, one should keep some primary characteristics of the cloud systems into mind which covers:-

- Pay per use[1] of the services (PPUS)
- Rapid elasticity concerns (REC)
- Location independent resource pooling (LIRP)
- Ubiquitous network access (UNA)
- On-demand self service (ODSS)

The various cloud resources are used by millions of users through pushing their tasks to the associated cloud network. It is a very much big challenge to manage and schedule these lots of tasks properly and efficiently within the cloud network. To do this efficiently and properly, various scheduling approaches are illustrated in [13], [14], [15], [16], [2], [17], [18], [19], and [10].

These scheduling approaches consider on many factors such as cost matrices (CMs) which are provided by credit of tasks (CoT) for a dedicated resource [13], a light weighted VM backfill scheduler based on Backfill approach to deliver tasks and a Meta-Task-scheduler based on QoS [14], essential needs of QoS [15] and [16], cloud network diversity, heterogeneity and its over-headed workload [18].

In the cloud network, the assignment of resources and respective task scheduling give the number of cloud computing systems required for decreasing the total cost. The problems, associated with the resource allocation are spontaneously come out due to the highly dynamic behavior of cloud working, as the availability of cloud servers is shown while the customers request for it at the same duration of time. Thus this analysis pays the attention on various scheduling algorithms within the cloud networks after considering above discussed issues, problems, strategies, characteristics, behavior, hidden aspects and other challenges. For the IaaS cloud networks, it's a critical process to schedule the cloud network resources. VMs are generally taken as scheduling entities in these IaaS network clouds which have to be assigned to physical cloud resources of the quite heterogeneous type of nature. In the cloud networks the dynamic selection of target cloud resources takes place in different-different but like ways. It may be selected either randomly at run time or by any other means [Round Robin (RR) or Greedy Approach i.e. waiting time (WT) and resource processing power (RPP) based]. The scheduling process of the task selection may be done on the Shortest Job First (SJF) basis, Coarse Grained Task Grouping (CGTG) basis, First Come First Serve (FCFS) basis or Priority Based Scheduling (PBS) etc. The scheduling algorithms are fully responsible for both- the 'selection of the executable tasks' and the 'corresponding cloud resources' which are to be allocated and mapped for the successful execution of the tasks. As each and every new successive selection strategy focuses on the approach that it will result a better output and eliminate or minimize the earlier algorithms' limitations and drawbacks.

Greedy Allocation:

Resource allocation algorithms (RAA) having greedy approach are very much suitable for those heterogeneous cloud resource environments which are quite dynamic in behavior and are connected to a process scheduler through a simple homogeneous environment of cloud communication, is described in [12]. Greedy approach for optimized profit is one of the best approaches that are used to determine the problems of task scheduling.

Algorithm: Greedy Resource Allocation Algorithm

```
Inputs: VirtualMachine
Outputs: outer ResID; B
Begin
set resid1=sleepresid1= -1, execute= 0;
//find the best 'resource'
for each resource in ResourceCache & not done
do if(resource is in (Suspended State or Waking State)
and resid1==-1 ) then
 find remaining Capacity of resource and check
if remaning resource capacity >0 then
resid1=resurceid.execute++
end
end if(resource is in sleeping State) and resid1==-1)
thenfind remaining Capacity of resource and
Check
  if remaning resource capacity >0
thensleepresid1=resurceid1;
end
   end
end
if(resid1== -1 & sleepres id1==-1) then
outResID=-1 , return outResId
end
if(resid1== -1) then get resource from resource cache
having res id
outResID=res id
end if(sleepresid1==-1) then
get resource from resource cache having
sleepresid outResID=sleepresid1
end if resource is sleeps ate then
power Up resource
end return outResID end
```

3.2 SOLUTION OF THE TASK SECHEDULING PROBLEMS

Now this paper shows the interest on developing an effective and optimized algorithm which can help to optimize the quality parameters related to the performance of the cloud system. After looking all above mentioned quality parameters, this paper would try to optimize the cloud system performance and the clients would see the major improvement in the cloud system's performance.

3.2.1. Deployment of the Activity Based Costing (ABC) Algorithm in Cloud Networks

Deployment of ABC algorithm can be illustrated by representing it in a parent-child relationship (PCR) structure (tree structure) which is shown in figure 3.

Implementing First Come First Serve (FIFO) approach i.e. according to the tasks' approaching time, the tasks would be resided in one parent queue. After it the called resources and data for their associated tasks would be traced, checked and arranged into two disjoint queues, first 'available' and second 'partially available'. In the first disjoint queue 'available queue', the inter-dependency and/or independency of the tasks would be traced, according to that again they would be placed in associated queues. Then in the second queue which is said as 'partially available' queue, there are already some tasks which may require for other data centers (DC) to deliver them other data resources. After it these tasks are sorted in various queue sub categories such as CT_1, CT_2, CT_3....and so on. Finally these queue sub-categories are carried out on the basis of arriving tasks' cloud resources, which they would need. For e.g., if there exists three different tasks- $task_1$, $task_2$ and $task_4$ and these three different tasks require cloud resources from two different locations of cloud data centers- DCL_1 and DCL_2, in this case these all three different tasks would be stored in one category. Let this category is CT_1. Similarly, the sub-

categories CT_2, CT_3…. and so on, would be made for other same type of tasks, until the second partially available queue gets fully empty.

Now, there are some major queues which are based on the dependencies and independencies of arriving tasks and CT_1, CT_2…….CT_N. So again three different priority-queues are designed on the basis of priorities High, Mid & Low for each and every major queue that are clearly explained above. Now this question will definitely arise that how their priorities would be decided?

The priorities would be decided by taking consideration of the four key factors- completing time of tasks, cloud resources, space needed for execution and profit associated with them.
The following given formula is derived to assign the priorities of arriving taskes:-

$$K_i = \sum_{j=0}^{n} (T_{i,j} + S_{i,j} + C_{i,j}) / P_i$$

where:

K_i : *priority of the* i^{th} *task*
$T_{i,j}$: *time required to complete* j^{th} *activity of* i^{th} *task*
$S_{i,j}$: *space needed to operate* j^{th} *activity of* i^{th} *task*
$C_{i,j}$: *cost of* j^{th} *activity in terms of resources of* i^{th} *task*
P_i : *profit from complete* i^{th} *task*
n : *total number of activities of any* i^{th} *task*

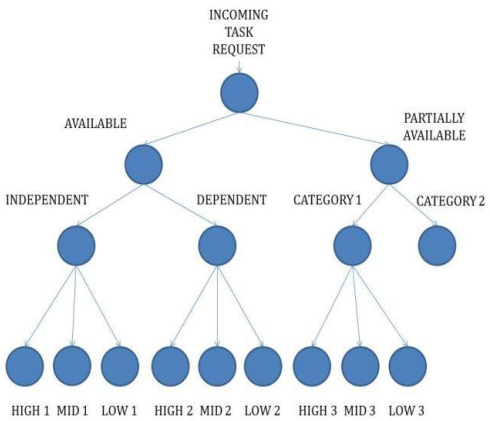

Figure 3. The tree structure of task scheduling

While determining the priorities of the arriving tasks, the newly assigned priority is measured with prior ones and after it the tasks are assigned into the already made priority-queues. When the assigning of tasks into the priority-queues gets over, then the assigned tasks from the High priority queues are chosen for their proper and efficient execution. After the successful execution of the queue-tasks from the top priority queues, the tasks of medium priority queues are transferred to the queues with top priority, and by like way the execution of the remaining queued tasks from all the filled queues gets initiated.

Further one important question arises again that how this selection of the queued tasks is sequenced in above mentioned way, as there are so many top priority queues? The answer of this question can be explained by a simple logic which is used in following manner:-

- Compare the different priorities of those all queued tasks which are queued at top priority level under their respective top priority queues
- After it chose the queued tasks having top most priority and assign the space and cloud network resources.
- If there are still remaining more resources, then select the next queued tasks considering same above mentioned strategy and assign their space and cloud resources.
- Now repeat all the above steps until the ending of all remaining cloud resources.
- When the assignments of all the cloud resources gets full with their capacity, then keep waiting until the completion of any task and as any of them gets completed, again choose next queued task and assign it to the free resource and space.

3.2.3 Algorithm for the Activity Based Costing (ABC)

```
for all tasks do
        store in a queue parent
end for
for every i^th task do
        check if all the resources available or not
        if yes
        move to queue available
        else
        move to queue p_available
end for
for all tasks at queue available do
        check if dependent
        if yes
        move to queue dependent
        else move to queue independent
end for

for all tasks at queue p_available do
calculate priority Ki
end for
for every Ki do
  put task in appropriate queues of priority
    High, Mid and Low
end for
compare High1j, High2j, High3j.....
.....High(2+N)j
select task with highest priority for execution
while system is running task do
   check if new task is available
   if yes
   calculate priority and place at appropriate
   queue
else continue
end if
scan queues to modify priority
if queues are not empty
select new task of highest priority
 else
 wait for new task to arrive
end if
end while
```

4.ANALYSIS OF THE PROPOSED ALGORITHM

4.1 Performance of the algorithm with respect to execution cost (EC)

The tasks chose the appropriate and proper resources after applying the greedy algorithmic approach and also minimize their execution cost (EC) separately. By the above proposed algorithm, the efficient execution of tasks outputs better than traditional sequential approach which is shown in figure 4. As the cloudlets get increased, this improvement in execution cost also gets increased-

No. Of Cloudlets	Proposed Algorithm	Sequential Assignment
25	565.91	735.68
50	1131.82	1471.36
75	1697.73	2207.05
100	2263.6	2942.73

Figure 4: Execution cost (EC): A relative comparison

The following graph in figure 5 is representing the relative comparison between the execution cost of the sequential algorithm and the above mentioned proposed cloud scheduling algorithms. The tasks and their execution costs are taken on two different axes (x-axis and y-axis) in following graph respectively:-

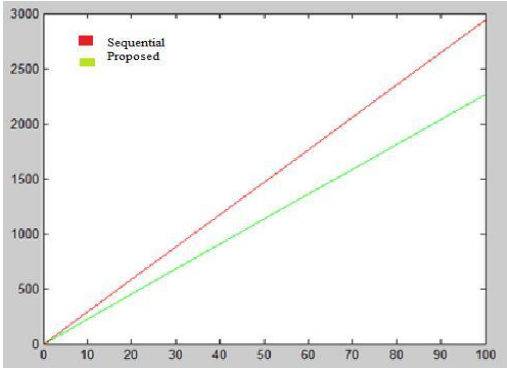

Figure 5: Execution Cost (EC): An analysis

4.2 Performance parameters with respect to finishing deadline

The following results represent the performance of the cloud network in each phase of trial as the cloudlets get increased:-

No. of Cloudlets	Proposed Algorithm	Sequential Approach
25	52.97	58.97
50	164.66	181.62
75	334.49	399.90
100	584.68	654.03
125	910.04	997.99
150	1298.50	1439.75

Figure 6: Relative Comparison between Completion Times (CTs)

The following graph in figure 7 is representing the relative comparison between the finishing times of the prior sequential and above mentioned proposed cloud scheduling algorithms. The tasks and their completion times are taken on two different axes (x-axis and y-axis) in following graph respectively:-

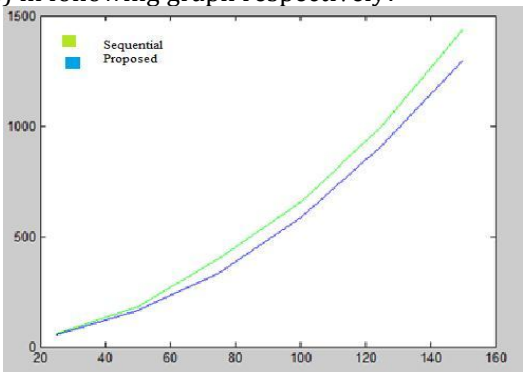

Figure 7: Completion Time (CT): An analysis

5. CONCLUSION

This paper tells the new algorithmic approach of task scheduling strategies (TSS) and resource allocation scheduling (RSA) in cloud networks. It also helps in the better understanding of cloud resource assignments and the task scheduling techniques. With the help of task scheduling techniques, the cloud resources of the cloud networks are effectively efficiently managed, assigned and optimized.

6. FUTURE SCOPE

The scheduling policies are always required for the computational benefits of the cloud clients and the cloud service providers both. Each new strategy, which can output the better result than prior strategies, would always be welcome in research world. As far as the future scope of this research work is concerned, the cost based scheduling (CBS) policy can be explained more efficiently after applying this approach.

7. REFERENCES

[1] QI CAO, ZHI-BO WEI, WEN-MAO GONG International School of Software Wuhan University Wuhan, China,". An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing", IEEE 2009.

[2] M. Malathi, "Cloud Computing Concepts", IEEE 2011.

[3] Fei Teng, "Resource allocation and scheduling models for cloud computing", Paris, 2011.

[4] Han Zhao, Xiaolin Li, "AuctionNet: Market oriented task scheduling in heterogeneous distributed environments", IEEE, 2010.

[5] Boloor, K., Chirkova, R., Salo, T., Viniotis, Y., "Heuristic-Based Request Scheduling Subject to a Percentile Response Time SLA in a Distributed Cloud". IEEE, 2011.

[6] Zaman S., Grosu D., "Combinatorial Auction- Based Dynamic VM Provisioning and Allocation in Clouds", IEEE, 2012

[7] Hao Li, Huixi Li, "A Research of Resource Scheduling Strategy for Cloud Computing Based on Pareto Optimality M×N Production Model",, IEEE, 2011.

[8] Zhongyuan Lee, Ying Wang, Wen Zhou, "A dynamic priority scheduling algorithm on service request scheduling in cloud computing", IEEE, 2011.

[9] Laiping Zhao, Yizhi Ren, Yang Xiang, Sakurai, K., "Fault-tolerant scheduling with dynamic number of replicas in heterogeneous systems", IEEE, 2011.

[10] Qi Cao, Zhi-Bo Wei, Wen-Mao Gong, "An Optimized Algorithm for Task Scheduling Based on Activity Based Costing in Cloud Computing", IEEE, 2009.

[11] Luqun Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers", IEEE, 2009.

[12] Qi Zhang, Quanyan Zhu, Boutaba, R., "Dynamic Resource Allocation for Spot Markets in Cloud Computing Environments", IEEE, 2012.

[13] Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao, NPU HPC Center Xi'an, China "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", IEEE 2010.

[14] Jeyarani, R., Ram, R. Vasanth, Nagaveni, N., "Design and Implementation of an Efficient Two-Level Scheduler for Cloud Computing Environment", IEEE, 2010.

[15] GAN Guo-ning, HUANG Ting-Iei, GAO Shuai School of Computer science and engineering Guilin University of Electronic Technology Guilin, China "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment", IEEE 2010.

[16] Huang Qi-yi, Huang Ting-lei, "An optimistic job scheduling strategy based on QoS for Cloud Computing", IEEE, 2010.

[17] Paul, M., Sanyal, G., "Survey and analysis of optimal scheduling strategies in cloud environment", IEEE, 2012

[18] Meng Xu, Lizhen Cui, Haiyang Wang, Yanbing Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing", IEEE, 2009.

[19] Kuan-Rong Lee, Meng-Hsuan Fu, Yau-Hwang Kuo, "A hierarchical scheduling strategy for the composition of Algorithms for Preemptable Job Scheduling in Cloud Systems", IEEE, 2010.

[20] Wei Wang, Guosun Zeng, "Trusted Dynamic Scheduling for Large-Scale Parallel Distributed Systems", IEEE, 2011.