Multiple Request Resource Allocation by using Time-Shared Policy in Cloud Computing

Nancy Goel
NCCE, Israna Panipat, India.

Sukhvir Singh
NCCE, Israna Panipat, India.

Abstract – Cloud computing is the enactment of using a network of remote servers hosted on the internet to accumulate, manage and process data rather than a local server or a personal computer. Cloud computing is known as a popular and important term in the IT society these days. As the main goal of cloud computing we can mention the better use of distributed resources and applying them to attain a higher throughput, performance and solving large scale computing problems. In this thesis we focus on allocation of resources on multiple request simultaneously, As we observe in previous work that the researcher found difficulty in allocation of resources to many requests as they faced many problems like regarding cost, time, bandwidth etc which affects the overall performance of the cloud therefore, In our proposed work we mainly focus on time consuming problem by using Scheduling algorithms in which we are using Time-shared Policy to get the appropriate results without any delay.

Index Terms – Cloud Computing, Resource Allocation, Round Robin, Time- Shared Policy.

1. INTRODUCTION

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on -demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in either privately owned or third-party data centers that may be located far from the user–ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over an electricity network. As well, it enables organizations to focus on their core businesses instead of spending time and money on computer infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables information technology (IT) teams to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This will lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.

Figure 1: Cloud Computing

In 2009, the availability of high-capacity networks, low-cost computers and storage devices as well as the widespread adoption of hardware virtualization, service-oriented architecture, and autonomic and utility computing led to a growth in cloud computing. Companies can scale up as computing needs increase and then scale down again as demands decrease.

In 2013, it was reported that cloud computing had become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. Some cloud vendors are experiencing growth rates of 50% per year, but being still in a stage of infancy, it has pitfalls that need to be addressed to make cloud computing services more reliable and user friendly.

2. PROBLEM STATEMENT

Cloud computing emerges as a new computing paradigm which aims to provide reliable, customized and QoS (Quality of Service) guaranteed computing dynamic environments for end-users. There are numerous advantages of cloud computing, the most basic ones being lower costs, re-provisioning of resources and remote accessibility. Cloud computing lowers cost by

avoiding the capital expenditure by the company in renting the physical infrastructure from a third party provider. Due to the flexible nature of cloud computing, we can quickly access more resources from cloud providers when we need to expand our business. The remote accessibility enables us to access the cloud services from anywhere at any time. To gain the maximum degree of the above mentioned benefits, the services offered in terms of resources should be allocated optimally to the applications running in the cloud.

3. PROPOSED WORK

In this thesis, we propose effective allocation scheme with the scheduling algorithm to ensure the allocation of resources to the customers in the cloud. We rely on providing the resources to all customers according to their demand in a less waiting time as compared to already existed algorithm's waiting time. This construction drastically reduces the waiting time and improves the performance of whole procedure. By using Modified Round Robin Algorithm which was earlier works on Space- Shared environment is now modified in a new way according to which now this modified Round Robin Algorithm works on Time-Shared Algorithm. As when the group of tasks (Cloudlet) sent by the users reaches to the Broker then this broker will manage all the tasks according to their need of resources then send them to the Datacenters where many of virtual machines are present, then they allocate the required resources to the users or we can say complete the tasks. After completion they send the results back to the Broker and then to customers.

Compared to many of the existed algorithm, which provides this service but with more waiting time, more bandwidth, more cost, the challenge- provide all these services but in less waiting time, to speed-up the whole procedure.

Unlike most prior works for ensuring optimal scheduling model, the new scheme supports the dynamic resource allocation including quality of service.

Improved performance and fast procedure shows that the proposed scheme is highly efficient and successfully allocated the resources dynamically.

3.1. SCHEDULING

The Job management is that the primary idea of cloud computing systems task scheduling problems are main that relates to the efficiency of the complete cloud computing system. Job scheduling may be a mapping mechanism from users' tasks to the correct choice of resources and its execution. Job scheduling is flexible and convenient. Jobs and job streams can be scheduled to run whenever required, based on business functions, needs, and priorities. Job streams and processes can set up daily or weekly or monthly and yearly in advance, and run on demand jobs without need for help from support workers.

3.2. ROUND – ROBIN ALGORITHM

It is one of the oldest, simplest, fairest and most widely used scheduling algorithms, designed especially for time-sharing systems. A small unit of time called time slice or quantum11 is defined. All runable processes are kept in a circular queue. The CPU scheduler goes around this queue, allocating the CPU to each process for a time interval of one quantum. New processes are added to the tail of the queue. The CPU scheduler picks the first process from the queue, sets a timer to interrupt after one quantum, and dispatches the process. If the process is still running at the end of the quantum, the CPU is pre-empted and the process is added to the tail of the queue. If the process finishes before the end of the quantum, the process itself releases the CPU voluntarily. In either case, the CPU scheduler assigns the CPU to the next process in the ready queue. Every time a process is granted the CPU, a context switch occurs, which adds overhead to the process execution time.

3.3. TIME – SHARED POLICY

In Time-Shared scheduling policy it schedules all tasks on virtual machine at the same time. It shared the time among all tasks and schedule simultaneously on the virtual machine. This policy is also used to schedule the virtual machine on the host. The concept of round-robin (RR) scheduling algorithm is used in this policy.

STEPS TO DEFINE TIME-SHARED POLICY

Step 1:- All accepted task are arranged under the queue.

Step 2:- Then schedule the task simultaneously on the virtual machine.

Step 3:- When queue is empty it checks for new task.

Step 4:- If new task arrives it schedule similarly as in the step 2.

Step 5:- End.

3.4. ALLOCATION OF RESOURCES

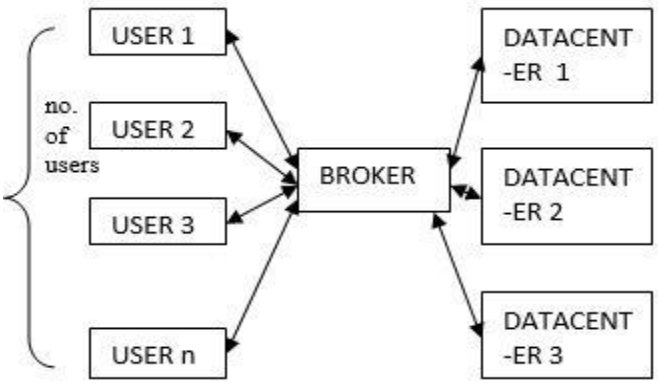

Figure 3.4: Basic Model for Allocation of Resources

In cloud computing, Resource Allocation (RA) is the process of assigning available resources to the needed cloud applications over the internet. Resource allocation starves services if the allocation is not managed precisely. Resource provisioning solves that problem by allowing the service providers to manage the resources for each individual module.

3.5. PROPOSED ALGORITHM

Algorithm Dynamic Allocation: (Selection of Cloudlet for execution of high priority Jobs)

Input: Cloudlets in execution; New high priority Cloudlet; Threshold

Output: Selected Cloudlet for execution of New high priority Cloudlet

Begin

For each Cloudlet in Cloudlets in execution

if (lease = suspendable || cancellable)

candidateSet.add(Cloudlet)

end if

end for

For each Cloudlet in candidateSet

if (Cloudlet.deadline < New high priority Cloudlet.deadline)

candidateSet.remove(Cloudlet)

end if

end for

For each Cloudlet in candidateSet

if(Cloudlet.execution > Threshold)

candidateSet.remove(Cloudlet)

end if

end for

if (candidateSet.count > 1)

Selected ← Cloudlet candidateSet.select(top)

end if

End

4. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

4.1. GENERATION OF TASKS

Resource Allocation starves services if the allocation in not managed precisely. As in this thesis, we have two ends first we have users at one side that generates the various no. of tasks which we send to the Datacenters with the help of Broker randomly. And this process of sending requests from the users to Datacenters to allocate resources is called Resource Allocation. In this way we would say that resources were demanded by the users and this demand of users will be fulfill at Datacenters. Where we have resources like Physical Entities, Bandwidth, Virtual Machines (VM), RAM, Host which we have different for every Datacenter. And this process of sending multiple requests is called Mapping of Tasks.

4.2. HANDLING OF TASKS

When Users generates the tasks and send them to the Broker, then after this, it is the duty of the Broker to handle those multiple requests or tasks & forward them for the allocation of the resources at Datacenters. It depends on the Broker that How it handles the requests at a time. Because Broker is the only one who can save the time or reduces the waiting time to do work efficiently As firstly all the requests reaches to the Broker then it manages all the tasks by sending them to the appropriate Datacenters where they will consumes less time as lesser time will be taken by the Datacenters more will be the speed of the whole process then Datacenters complete their work and then return the output to the Broker as soon as possible and then to users itself, in this way Quality of Service (QoS) will be improved.

4.3. SIMULATION

When the Broker handles the tasks and send them to the Datacenters according to their time quanta, then after reaching to the Datacenter it boot up the Virtual Machine (VM) and give required RAM, Bandwidth to the Virtual Machine in this way the virtual machines can start their work. As in this proposed work we are using Time-Shared Scheduling Policy which schedules all tasks on virtual machine at the same time. It shared the time among all tasks and schedule simultaneously on the virtual machine. For example: If VM set the time 5 sec for each task then it execute the tasks one by one in their fixed quanta. In this way the Allocation of resources demanded by multiple users will takes place at the same time. After this when the queue will empty then it checks for the new tasks.

4.4. MIGRATION FOR ALLOCATION

As the tasks reaches to the Datacenters for execution then as we know we are following Time-Shared Scheduling Policy in our proposed work so VM shared the time in the form of MIPS

(Micro Instruction Per Second) along all tasks and schedule simultaneously and the time consumed by the Datacenter continuously get noticed by the Broker as if we have time quantum of 5 MIPS then as every task is comprised of instruction then for ex. If first task has 5 instruction and each instruction takes 1 MIPS for execution then according to fixed quantum that task will be completed in exactly 5 MIPS, Now the second task has 2 instruction then it will take 2 MIPS for execution but as we know we have time quantum of 5 MIPS per task so we have 3 MIPS which are remaining from the second task, that can be used for the next task and this is known as the Migration.

4.5. FLOW CHART

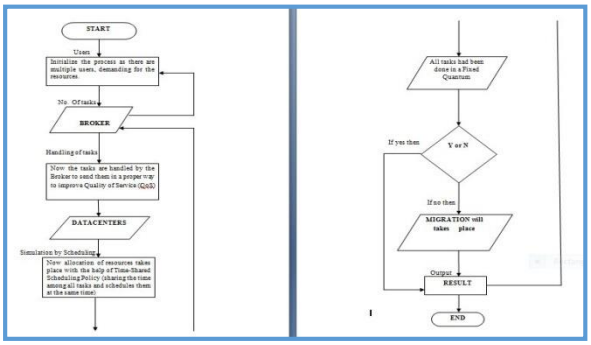

Figure 4.5: Flowchart

5. RESULTS

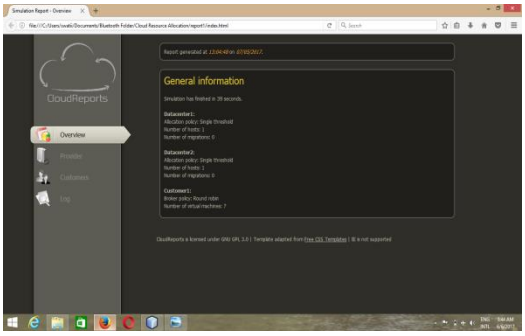

Figure 5(a): First Screen after running the code

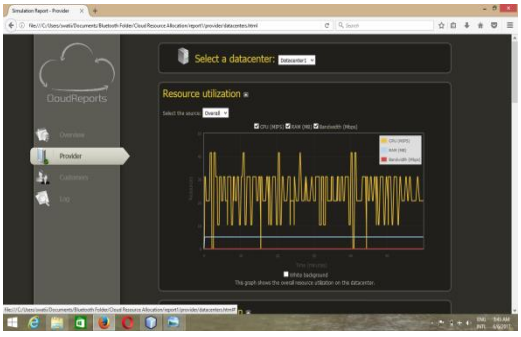

Figure 5(b): Overall resource utilization by Provider

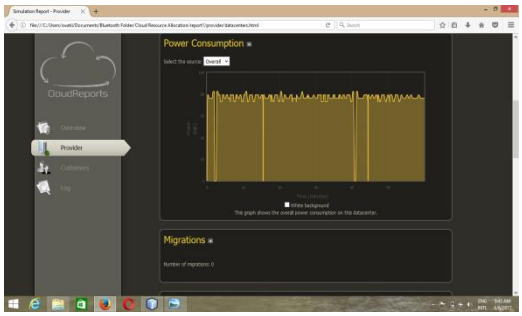

Figure 5(c): Overall Power consumption by Provider

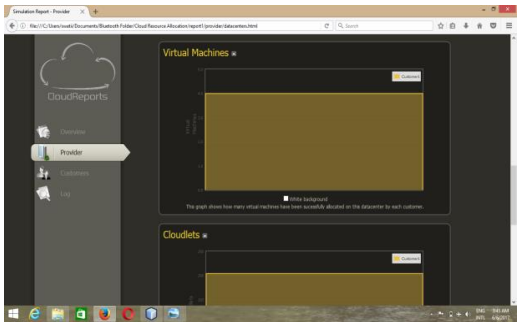

Figure 5(d): Successfully allocated VM by Customers

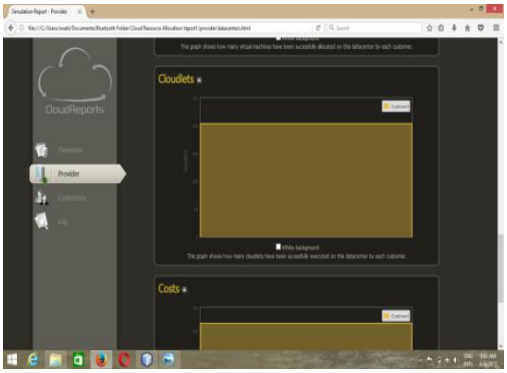

Figure 5(e): Successfully executed Cloudlets by Customers.

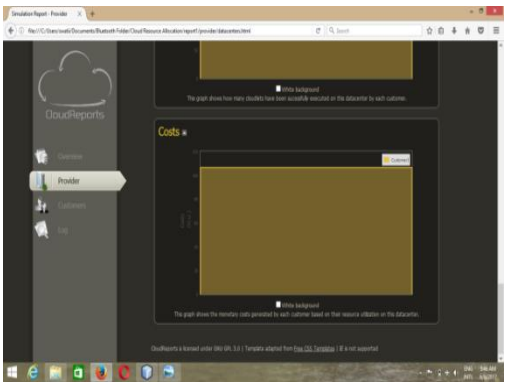

Figure 5(f): Cost generated by Customers..

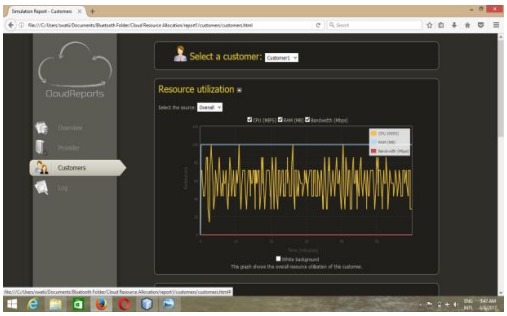

Figure 5(g): Resource utilization of this Customer

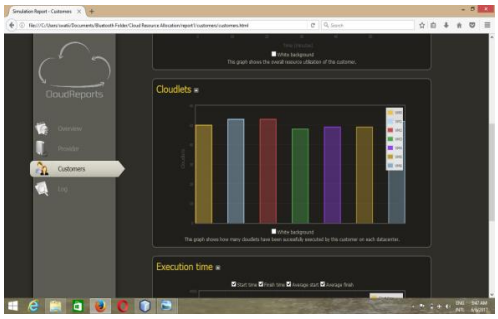

Figure 5(h): Successfully executed Cloudlets by this Customer

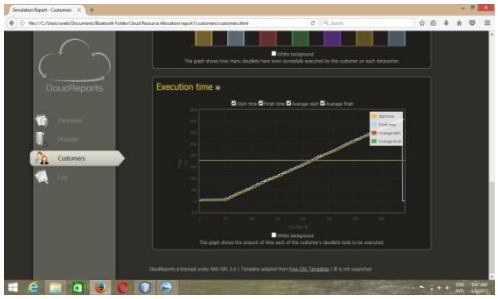

Figure 5(i): Execution time of this Customer

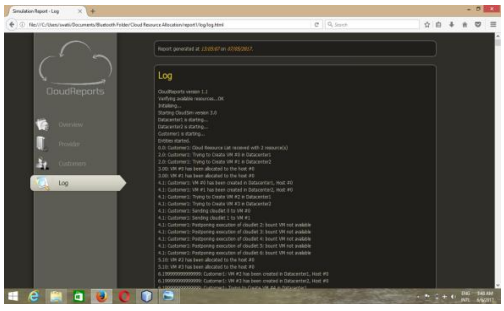

Figure 5(j): Log File

6. CONCLUSION

In this thesis, we investigated the problem of multiple request resource allocation in cloud computing. To ensure the allocation of resources in cloud we proposed an effective algorithm, through which whenever more than one user sends request for resources to datacenters then before this proposed algorithm, this procedure was so time consuming, but now as we proposed effective and dynamic algorithm by tuned Timeshared Policy to Round- Robin Algorithm, and it gives better results, we used Modified Round Robin(Space- Shared Policy) as a reference, it is concluded that for allocation of resources Time-Shared Policy tuned Round Robin Algorithm gives better results in comparison to Space-Shared Policy.

7. FUTURE SCOPE

We believe that Allocation of resources on multiple requests in cloud computing, an area full of challenges, is still in its infancy now, and many research problems are yet to be identified. As an interesting question in this model is, if we can construct a model to de-allocate the resources.

REFERENCES

[1] V.Vinothina, Dr.R.Sridaran, and Dr.Padmavathi Ganapathi, "A Survey on Resource Allocation strategies in Cloud Computing," *International Journal of Advanced Computer Science and Applications(IJACSA)*,vol. 3 no. 6, pp. 97-104, 2012.

[2] D. Prasad, Fathima Zehra, S.V.Hemanth, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," *International Journal of Computer Science And Technology(IJCST)*,vol. , issue 2, pp. 610-622, June 2012.

[3] Venkatesa Kumar, V. and S. Palaniswami, "A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing,"*Journal of Computer Science*, pp. 780-788, 2012

[4] Amit Kothari, Nootan Padia, and Maulik Parekh, "Real Time Resource Allocation Methodology and Quality based Framework in Cloud Computing," *International Journal of Computer Application(IJCA)*,vol. 62 no. 11, pp. 31-34, January 2013.

[5] Rupali Shelke, and Rakesh Rajani, "Dynamic Resource Allocation in Cloud Computing," *International Journal of Engineering Research and Technology (IJERT)*,vol. 2 issue 10, pp. 843-846, October 2013.

[6] B. Karthikeyan, T. Sasikala , and M. DilliBabu, "Dynamic Resource Allocation for Cloud Computing including the feasibility study on Green Computing," *International Journal of Science and Research* (*IJSR*), vol. 3 issue 6, pp. 1230-1232, June 2014.

[7] Vinayak Awasare, and Sudarshan Deshmukh, "Survey and Comparative Study on Resource Allocation Strategies in Cloud Computing Environment," *IOSR Journal of Computer Engineering (IOSR-JCE)*,vol. 16 issue 2, pp. 94 - 101, April 2014.

[8] Priyanka Mod, and Mayaank Bhatt, "A Survey on Resource Allocation Technique in Cloud Computing," *International Journal of Advanced Research in Computer and Communication Engineering(IJARCCE)*,vol. 3, issue 8, pp. 7815-7818, August 2014.

[9] Saima Israil, Rajeev Pandey, and Uday Chorasia, "Dynamic Resource Allocation for Overload Avoidance and Green Cloud Computing," *International Journal of Computer Applications (IJCA)*, vol. 126, no. 12, pp. 6-13, September 2015.

[10] Abhijits inh Parmar, and Rutvik Mehta, "An Extensive Analysis on Resource Allocation in Cloud Computing Environment," *International Journal of Advance Engineering and Research Development (IJAERD)*, vol. 2 issue 5, pp. 303-308, May 2015.

[11] Y. Bharath. Bhushan, and V. Bhavani, "Dynamic Resource Allocation using Virtual Machines and Parallel Data Processing in Cloud," *An International Journal of Advanced Computer Technology*, vol. 4 issue 11, pp. 2001-2005, November 2015.

[12] Ashwini. E, "Dynamic Allocation of Virtual Machines in Cloud Computing," *Global Journal for Research Analysis (GJRA)*,vol. 5 issue 5, pp. 266-268, May 2016.

[13] Vinayak Sontakke, Paresh Patil, Sumedh Waghamare, Rajani Kulkarni, N. S. Patil, and M. Saravanapriya, "Dynamic Resource Allocation Strategy for Cloud Computing using Virtual Machine Environment," *International Journal of Engineering Science and Computing (*IJESC*),*vol. 6 issue 5, pp. 4804 – 4806,May 2016.

[14] C. Vijaya, DR. P. Srinivasan, "A Survey on Resource Scheduling in Cloud Computing," *International Journal of Pharmacy and Technology (IJPT),* vol. 8 issue 4, pp. 26142-26162, November 2016.

[15] Pandaba Pradhan, B N B Roy, Prafulla ku. Behera, " Modified Round Robin Algorithm for Resource Allocation in Cloud Computing," *Procedia Computer Science,* 85 (2016) 878-890.

[16] Mr. S Natarjan, Dr. R Pugazendi, " Survey on Dynamic Resource Allocation Techniques in Cloud Computing," *International Journal of Computer Science and Mobile Computing,* vol. 3 issue 8, pp. 395-403, August- 2014.

[17] Vineet Khisty, Mrs. Jeba Priya, " A Survey on Dynamic Resource Allocation Techniques in Cloud Computing," *International Journal of Engineering Research and Technology,* vol. 3 issue 1, pp. 2555 – 2559, January- 2014.

[18] Kamini Bharti, Kamaljit Kaur, " A Survey on Resource Allocation Technique in Cloud Computing," *ird India,* vol. 3 issue 2, pp. 31 – 35, 2014.

[19] Pankaj Sareen Parveen Kumar, " Resource Allocation Strategies in Cloud Computing," *International Journal of Computer Science and Communication Networks,* vol. 5 issue 6, pp. 358-365, December 2015.

[20] B. Rajasekar, S.K Manigandan, " An Efficient Resource Allocation Strategies in Cloud Computing," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 3 issue 2, pp. 1239 – 1244, February 2015.

[21] V. Ram. Prabha, Ru. Va. Gayathri, S. Ebenzer. Juliet, " Overview of Dynamic Resource Allocation Strategies in Cloud Computing," *International Journal of Innovations and Advancements in Computer Science,* vol. 3 issue 12, pp. 42 – 48, December 2016.

[22] Shivani Sharma, Dhanshri Parihar, " A review on Resource Allocation in Cloud Computing," *International Journal of Advance Research , Ideas and Innovations in Technology,* vol. 1 issue 3, pp. 1 – 7, December 2014.

[23] N. Asha, Dr. G Raghavendra Rao, " A Review on Various Resource Allocation Strategies in Cloud Computing," *International Journal of Emerging Technology and Advanced Engineering,* vol. 3 issue 7, pp. 177-183, July 2013.

[24] M. Gokilavani, S. Selvi, C. Udhayakumar, " A Survey on Resource Allocation and Task Scheduling Algorithms in Cloud Computing," *International Journal of Engineering and Innovative Technology,* vol. 3 issue 4, pp. 173-179, October 2013.