EFFECTIVE MIGRATION RESOURCE REALLOCATION BASED ON CLOUD COMPUTING

Mahidhar Rao
Dept. Computer Science & Engineering
SRM Institute of Science and Technology
Chennai, India

Pavan Kumar Reddy
Dept. Computer Science & Engineering
SRM Institute of Science and Technology
Chennai, India

Harish Kumar
Dept. Computer Science & Engineering
SRM Institute of Science and Technology
Chennai, India.

Asst. prof. Ethirajulu V (Sr.G)
Dept. Computer Science & Engineering
SRM Institute of Science and Technology
Chennai, India

ABSTRACT:

A dynamic technology fundamentally transforming the way that computing services are delivered, cloud computing offers information and communication technology users a new dimension of convenience of resources, as services via the Internet. Because cloud provides a finite pool of virtualized on-demand resources, optimally scheduling them has become an essential and rewarding topic, where a trend of using Evolutionary Computation (EC) algorithms is emerging rapidly. Through analyzing the cloud computing architecture, this survey first presents taxonomy at two levels of scheduling cloud resources. It then paints a landscape of the scheduling problem and solutions. According to the taxonomy, a comprehensive survey of state-of-the-art approaches is presented systematically. Looking forward, challenges and potential future research directions are investigated and invited, including real-time scheduling, adaptive dynamic scheduling, large-scale scheduling, multi objective scheduling, and distributed and parallel scheduling .

INTRODUCTION:

Cloud computing is model one of the most enabling technologies to meet customer demand flexibly. Usually, it includes SaaS, PaaS, and IaaS. Software as a service (SaaS) provides access to complete applications as a service [1], and platform as a service (PaaS) provides a platform to develop other applications, such as the Google App Engine (GAE) [2]. Infrastructure as a Service (IaaS) [3–4] provides an environment to deploy the managed virtual machines. A reasonable resource allocation strategy can help to consolidate resources and reduce energy consumption. From the perspective of the providers, the key issue to be solved is to maximize the utilization by reducing the fundamental costs. As a core technique, virtualization [5–7] provides an effective way to pack the application requests into the VMs. The virtualization technique can make full use of the utilization by decreasing the power consumption. Virtual mapping [8] has become one of the core techniques in datacenters, which provides a solution to the resource allocation. Generally, the problems to be solved are divided into two subproblems: when to migrate and where to locate. Traditionally, researchers have focused more on energy consumption with the single objective of CPU utilization. The VM placement problem is usually solved by the bin-packing algorithm, which is an NP-hard problem [9–10]. For example, the pMapper system [11] proposed to determine the power-cost trade-offs by minimizing the costs with the minimum number of machines using the improved FFD algorithm. Another approach is best fit decreasing (BFD). Researchers proposed modified best fit decreasing (MBFD), which is an extension of the BFD method to improve energy efficiency under dynamic workloads [12]. However, these binpacking algorithms focus on improving the energy efficiency but ignore other elements, such as service level agreement (SLA) violation and resource wastage, which have impacts on the dynamic scheduling process. However, additional elements would make the bin-packing algorithm more complex. Most researchers focus on heuristic algorithms to solve dynamic scheduling problems. For example, the simulated annealing virtual machine placement (SAVMP) approach further improves of first fit (FF) algorithm, which minimizes the total power consumption in the datacenter [13]. It reduces the energy consumption based on CPU utilization; however, other factors should also be focused, such as the CPU, the memory. In [14], the genetic algorithm is used to reconfigure resources to minimize the migration cost, which has the advantage of proposing a weight function that includes the CPU and the memory. However, the migration cost function always takes the memory as the only optimization objective. The modified PSO (MPSO) algorithm [15] was introduced to improve the energy efficiency of the CPU and the disk. It makes use of the utilization and reduces the number of VM migrations. To avoid falling into a local search, several authors have proposed two-phase mechanisms to solve the optimization problem. In [16], the GA-ACO algorithm was proposed to improve the performance. The GA algorithm is used for the local search, and the ACO algorithm is used to escape the local search to improve the search performance. The proposed GA-ACO algorithm improved the performance effectively, but it considered fewer factors, such as the performance. In [17], the optimization model was used to minimize the total cost. The proposed ACO and GA algorithms are used in the global and local searches. However, from the perspective of the providers, more elements should be considered during the scheduling process, such as maximizing CPU utilization and memory utilization. In summary, many researchers have focused on improving energy efficiency [18–20]. However, few studies have investigated solving the multiobjective optimization problem. Providers should consider additional factors, such as reducing the power consumption, maximizing the utilization, and avoiding SLA violations. Hence, we present a two-level algorithm to achieve lower costs and power consumption.The first phase determines the hotspots by using a proposed score model and then migrates the VMs by using the PSO algorithm. The second phase finds the locations by using the improved ACO algorithm. Generally, the proposed algorithm aims to maximize the utilization and minimize the energy consumption. The contributions in this paper are as follows .

• First, the proposed method solves the issue of when to migrate. We propose a score model to determine the hotspots by using the GRA and TOPSIS methods, which simultaneously considers multiple objectives, such as CPU utilization and memory utilization .

• Second, this method migrates the VMs quickly. In this phase, we use the improved PSO algorithm to find the VMs to migrate by considering both CPU utilization and memory utilization. The PSO algorithm obtains the results quickly .

• Thirdly, this method solves the issue of where to migrate. It locates the suitable positions to reduce the rental costs,SLA violations and power consumption by using the consolidation strategy, which is implemented by the proposed Ant Colony Optimization (SACO) algorithm .

RELATED WORK

Dynamic scheduling strategy

Previous studies on resource consolidation strategies are divided into three main categories: static strategies [22–23], dynamic scheduling [24–26] and decision-making on the prediction [27–29]. The traditional static approach can be implemented to meet a varying demand, but it generates more overheads Halder et al. [30] presented a static consolidating algorithm that considered CPU utilization and SLA violations. Tiago et al. [31] proposed an LP and heuristics method to complete the mapping from the VMs to the host, which would reduce the number of migrations with the minimum penalty. The disadvantage of the static consolidation strategies is that they cause resource wastage to meet sudden load demands. An example of a dynamic scheduling strategy is the depending on the prediction technique. Cloudscale [32] achieved adaptive resource allocation with lower resource and energy costs by integrating VM resources with dynamic voltage and frequency scaling (DVFS) to save energy. Press [33] proposed the fine-grained mechanism, which reduces the resource wastage and SLO violations. However, the predictive technique is complex, and it is difficult to obtain accurate results with this technique. In dynamic scheduling strategies, the issue to be solved is when and where to migrate. Lova´sz et al. [34] presented a dynamic strategy that uses the greedy and modified first-fit algorithm and considers the power and response time as the performance metrics. Seyed et al. [35] proposed an adaptive threshold-based algorithm to detect overloaded hosts, which considered the optimization based on the energy performance trade-off. The best method is to determine the adaptive threshold by learning, but the disadvantage is that it focuses on the energy consumption; additional elements have effects during the dynamic scheduling process, such as the SLA and migration cost. Most studies have focused more on the single objective of minimizing energy consumption. However, more factors should be considered during the scheduling process. For example, providers also emphasize maximizing the utilization, including the CPU and the memory. In addition, to reduce SLA violations, the proposed two-level method reduces energy and resource wastage .

Multi objective optimization

One of the most important factors in server consolidation algorithms is the energy consumption. However, additional factors (e.g., the cost overhead, memory utilization, and SLA violations) should also be considered in the optimization algorithm. For example, Leili et al. [12] proposed an adaptive fuzzy threshold to detect overloaded or underloaded thresholds. The advantage of the method is that it proposes a double adaptive threshold to determine when the migration starts or where the VMs migrate. However, the proposed approach uses the energy and performance as the evaluation metrics. The MISTRAL [36] architecture proposed a strategy to reduce the power consumption and adaptation cost, which implemented the cost decision-making based on the response time. However, it developed a control architecture to solve the power trade-offs rather than several objectives. In [37], a multiple objective ant colony system algorithm that focuses on two objectives, including the makespan and the user's budget,was presented. This strategy had the advantage of minimizing the rental costs more efficiently; however, it focused more on maximizing the resource utilization. Additionally, it ignored other elements, such as the SLA and resource wastage. In [38], the genetic algorithm was proposed to enhance the system provisioning, system performances, system failure and network overheads. The method considered more factors than the previous methods, but it ignored the cost overheads. In [39], the CMCVRP optimization model was presented to reduce costs and energy consumption. The method used cost reduction model to achieve the cost reduction percentages. In [40], the proposed method allocated the resources to minimize the total amount of resources while meeting the end-to-end performance requirements for the application. The method described in [41] provided a theoretical control solution to the dynamic capacity provisioning problem that minimizes the total energy cost while meeting the performance objective of task scheduling delays. The response time and the temperature also play important roles in data centers, which can greatly affect the service quality. Based on the studies presented above. The main optimization parameters are the minimization of the energy overheads and the cost overhead and the maximization of the application performance. We propose a two-level hybrid algorithm. First, the algorithm determines when to migrate. The proposed score model based on the GRA and TOPSIS methods achieves the adaptive threshold depending on the CPU utilization and memory utilization. Second, the hybrid heuristic algorithm includes the PSO and ACO algorithms, which emphasizes reducing SLA violations and the energy consumption. In addition, the PSO algorithm determines the hotspot VMs quickly, and the proposed ACO algorithm focuses on solving the VM placement problem to reduce SLA violations and the energy consumption .

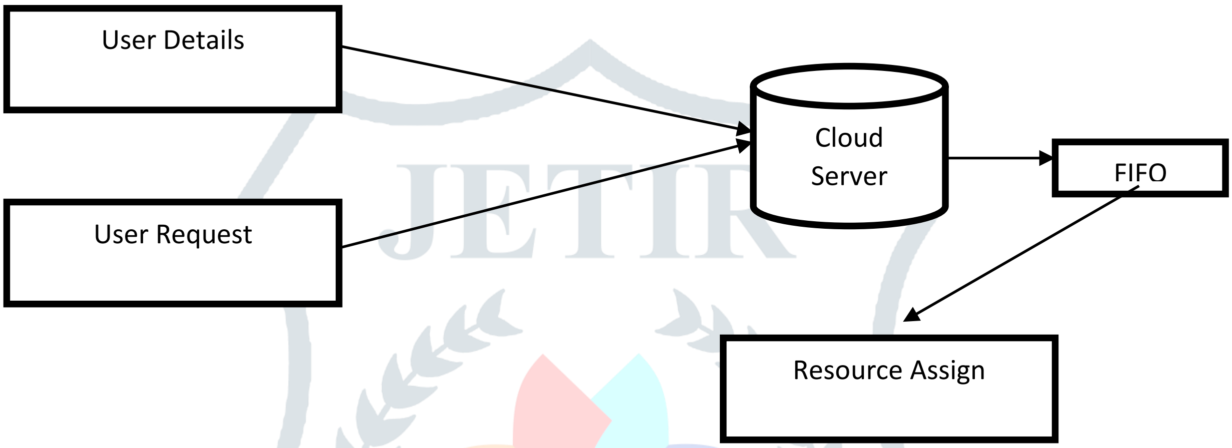
JA 19

User Registration

In this module we are going to create a User application in which the User is obtained to access the data from the Server of the Cloud Service Provider. Here first the User needs to create an account and then only they are allowed to access the Network. Once the User create an account, they are to login into their account and request the Job from the Cloud Service Provider. Based on the User's request, the Cloud Service Provider will process the User requested Job and respond to them. All the User details will be stored in the Database of the Cloud Service Provider. In this Project, we will design the User Interface Frame to Communicate with the Cloud Server through Network Coding using the programming Languages like Java/ .Net. By sending the request to Cloud Server Provider, the User can access the requested data if they authenticated by the Cloud Service Provider .

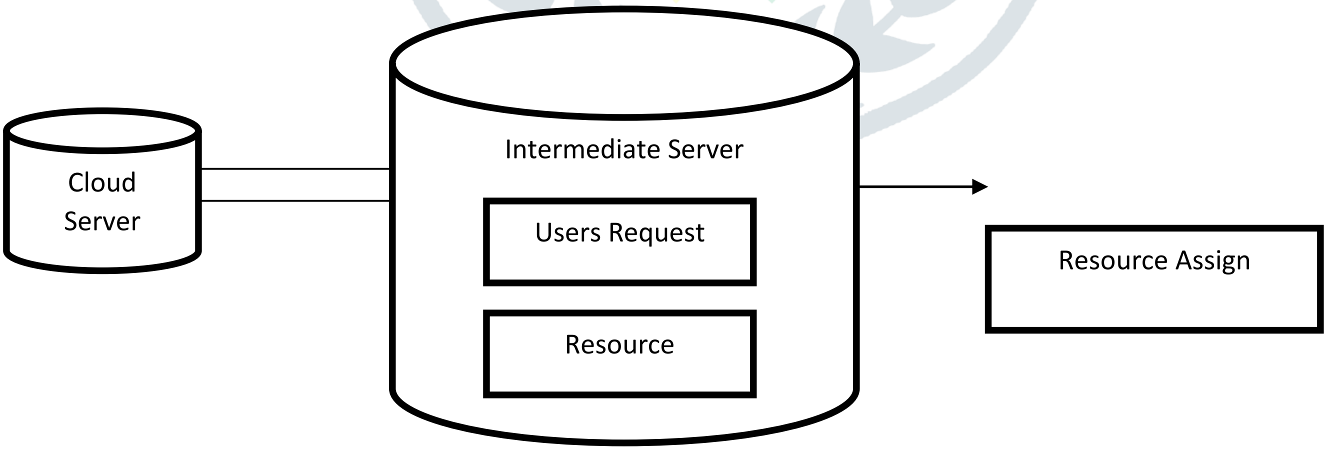
Cloud Server Deployment

Cloud Service Provider will contain the large amount of data in their Data Storage. Also the Cloud Service provider will maintain the all the User information to authenticate the User when are login into their account. The User information will be stored in the Database of the Cloud Service Provider. Also the Cloud Server will redirect the User requested job to the Resource Assigning Module to process the User requested Job. The Request of all the Users will process by the Resource Assigning Module. To communicate with the Client and the with the other modules of the Cloud Network, the Cloud Server will establish connection between them. For this Purpose we are going to create an User Interface Frame. Also the Cloud Service Provider will send the User Job request to the Resource Assign Module in Fist in First out (FIFO) manner .



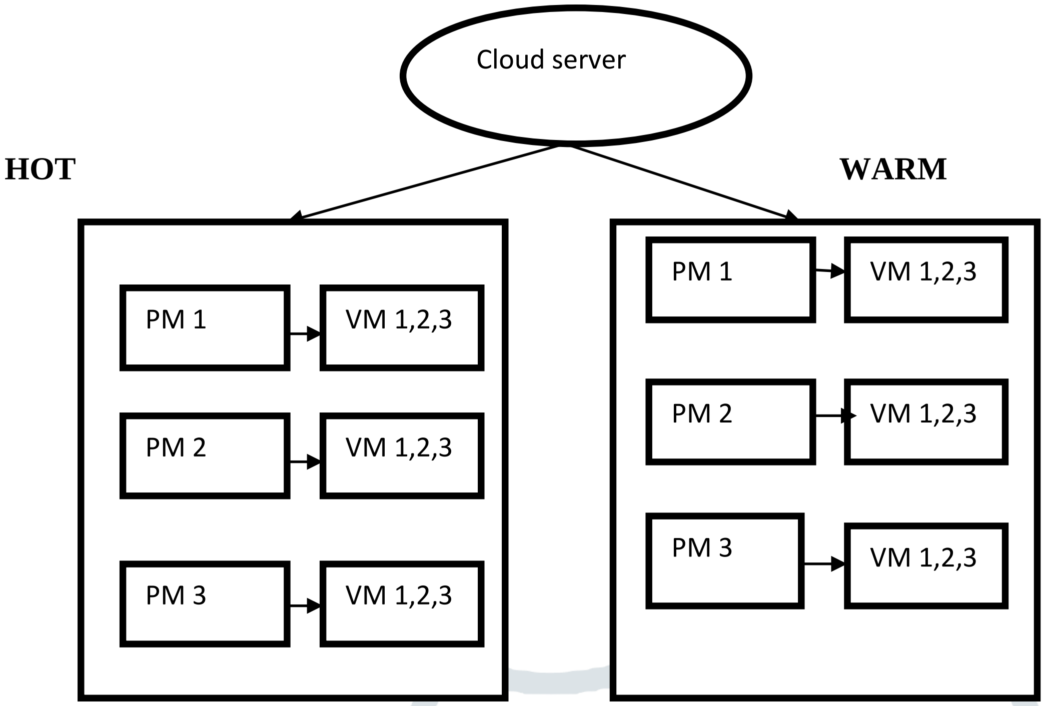
Intermediate Server Deployment

By implementing Intermediate Server the Job Processing Scheme, Systematically we can process the User Request Job and productively maintains the Resources of the Cloud Server. So that we can optimise the Energy of the Resources when they are not exercise the Job.



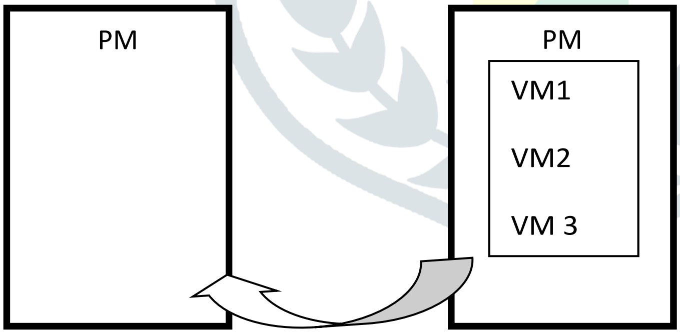
Green Computing Setup

Green computing is the term used to denote efficient use of resources in computing. It Is also known as Green IT . In this Module, we will Process the User requested Job. The User requested Job will redirect to the RAM of the Cloud Server. The RAM will contain two Types of the Physical Servers. 1. HOT Server and WARM Server These Physical Servers will contain 'n' number of virtual Server to process the User requested Job. So that the Job can be efficiently processed .



Migration Of Virtual Server

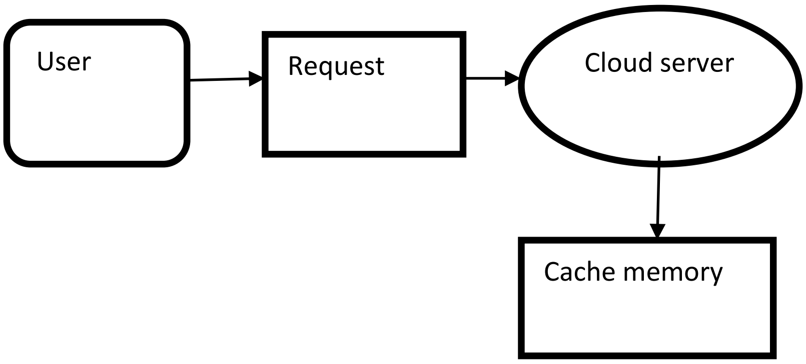
In this module we create the migration server, main use of migration to migrate the job form on virtual serve to another server, so that the energy can be reduce and work load of the server is balanced, by using the Migration we can shift the process from one VM to anther VM without loss of data.



MIGRATION

Cache Server Implementation

As a modification in this Project, we are creating a Cache Memory in the User requested job will be stored for the period time. If the another User requests the same Job to the Server of the Cloud Service Provider (CSP), the Server will check in the Cache Memory first . So that we can reduce the Job Processing Time. If the request Data is presented, then the Server will provide the Data to the User immediately. If the request Data is not in the Cache Memory, then the Server process the User requested Job by transferring it to the RAM .



Resource Allocation "Resource allocation involves deciding what, how many, where, and when to make the resource available to the user. Typically, users decide the type and amount of the resource containers to request then providers place the requested resource containers onto nodes in their datacenters. To run the application efficiently, the type of resource container need to be well matched to the workload characteristics, and the amount should be sufficient to meet the constraints i.e., job must be completed before its deadline. In an elastic environment like the Cloud where users can request or return resources dynamically, it is also important to consider when to make such adjustments."

Job Scheduling "Once the resource containers are given to the user, the application makes a scheduling decision. In many cases, the application consists of multiple jobs to which the allocated resources are given. The job scheduler is responsible for assigning preferred resources to a particular job so that the overall computing resources are utilized effectively. The application also has to make sure each job is given adequate amount of resources, or its fair share. Such a scheduling decision becomes more complex if the environment is heterogeneous."

Algorithm:

Allocation Process

1. Initialize & Start the servers on Cloud Datacenters

2. Accept ClientId & No. of CPUs requested.

3. Start the allocation process for client ClientId on First Server.

4. If remaining Capacity of server is more than the requested CPUs then

    (i) Place the request of ClientId on the server.

    (ii) Update the utilization Percentage and remaining capacity of server

    (iii) Set allocated status to true for the client ClientId.

    (iv) Exit.

Else

(i) Start the allocation process for client ClientId on Next Server.

(ii) Goto Step 4.

[end of If]

5. If no next server available then Write: “Datacenter is Out of Server”.

6. Exit.

CONCLUSION

Cloud computing is a technique to provide and maintain data and applications with the help of internet and remote servers. The presented work is focused on the concept of effective resource allocation, de-allocation and reallocation in a cloud environment. As the particular cloud will get the request, it will search for the number of requested processors. If the numbers of processors are available with the current cloud, the resources will be allocated to that particular client. But if the sufficient numbers of processors are not available then the search will be performed for the next particular cloud to perform the resource allocation. When the client stops the task then the service allocated to the client is released & same can be reallocated to another client in the waiting. We can also some migration work so that if a cloud is under utilize then its services are migrated to nearest cloud having sufficient utilization .

REFERENCES

[1] B. Hayes, “Cloud Computing,” Commun. ACM, vol. 51, no. 7, pp. 9–11, Jul. 2008.

[2] P. Mell and T. Grance, “The NIST definition of cloud computing (draft),” NIST special publication, vol. 800, no. 145, p. 7, 2011.

[3] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, “Cloud Computing, A Practical approach”

[4] D. Warneke, O. Kao, “Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud”, IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 6, pp 985 - 997, June 2011, DOI: http://doi.ieee computersociety.org/10.1109/TPDS.2011.65.

[5] V. Vinothina, Dr. R. Sridaran, Dr. Padmavathi Ganapathi, “Resource Allocation Strategies in Cloud Computing”, International Journal of Advanced Computer Science and Applications [IJACSA], Vol. 3, No.6, 2012. ISSN: 2158-107X (Print), DOI: 10.14569/issn.2156- 5570.

[6] Sowmya Koneru, V N Rajesh Uddandi, Satheesh Kavuri, ”Resource Allocation Method using Scheduling methods for Parallel Data Processing in Cloud”, International Journal of Computer Science and Information Technologies[IJCSIT], Vol. 3(4), 2012, pp 4625 - 4628 4625, ISSN: 0975-9646.

[7] Gihun Jungand, Kwang Mong Sim, "Location-Aware Dynamic Resource Allocation Model for Cloud Computing Environment", International Conference on Information and Computer applications (ICICA 2012), pp 37 - 41, IPCSIT, Vol. 24, IACSIT Press, Singapore.

[8] Thangaraj P, Soundarrajan S, Mythili A, "Resource allocation policy for IaaS in Cloud computing", International Journal of Computer Science and Management Research, Vol 2, Issue 2, pp 1645 - 1649, February 2013, ISSN 2278-733X.

[9] Mohd Hairy Mohamaddiah, Azizol Abdullah, Shamala Subramaniam, and Masnida Hussin, "A Survey on Resource Allocation and Monitoring in Cloud Computing", International Journal of Machine Learning and Computing, Vol. 4, No. 1, February 2014.

[10] Pratik P. Pandya, Hitesh A. Bheda, "Dynamic Resource Allocation Techniques in Cloud Computing", International Journal of Advance Research in Computer Science and Management Studies Volume 2, Issue 1, January 2014.

[11] Nguyen Trung Hieu, Mario Di Francesco, and Antti YlaJaaski, "A Virtual Machine Placement Algorithm for Balanced Resource Utilization in Cloud Data Centers", 2014 IEEE International Conference on Cloud Computing.

[12] 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, February 2015.

[13] Jin, D. Pan, J. Xu, and N. Pissinou, "Efficient VM placement with multiple deterministic and stochastic resources in data centers" in IEEE Global Communications Conference (GLOBECOM), 2012, pp. 2505–2510.

[14] Li, Bo, Jianxin Li, Jinpeng Huai, Tianyu Wo, Qin Li, and Liang Zhong. "Enacloud: An energy-saving application live placement approach for cloud computing environments." In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, pp. 17-24. IEEE, 2009.

[15] Kokilavani, T., Dr George Amalarethinam, "Load balanced min-min algorithm for static meta-task scheduling in grid computing", International Journal of Computer Applications 20, no. 2 (2011): 43-49.

[16] Sumita Bose, Jitender Kumar, "An Energy Aware Cloud Load Balancing Technique using Dynamic Placement of Virtualized Resources", Advances in Computer Science and Information Technology (ACSIT) Volume 2, Number 7; April – June, 2015 pp 81 – 86.