Volume 6, Issue 8, August 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improved the Performance of Load Balancing Technique Using Coupling of Virtual Machine

Shatrunjay Panday
M Tech Software Engineering,
RKDF Institute of Science & Technology, Bhopal, India

Prof. Daya Shankar Pandey
Assistant Professor,
RKDF Institute of Science & Technology, Bhopal, India

Prof. Shrikant Lade
RKDF Institute of Science & Technology, Bhopal,
India

Abstract: The limited resource and unlimited number of user always wanted the process of balancing condition. The process of load balancing improves the efficiency of cloud system environment. The process of cloud load balancing used various methods and algorithm for process of balancing. The dynamic load balancing technique used heuristic and queuing based searching and allocation technique for the balancing of job in cloud environment. In cloud resource overload impact the performance of cloud computing environment. Now the process of load balancing plays a major role in cloud computing environment. The various researcher and scientists used dynamic and static load balancing technique. in dynamic load balancing technique used heuristic function such as ACO, PSO, Genetic algorithm and many more guided searching algorithms. in this paper proposed graph based technique for the allocation of job in load balancing process. The graph based technique basically used the concept of sharing of virtual machine. the shared virtual machine allocated the job in dedicated time period for the execution of process.

Keyword: - ACO, PSO, GSO, Cloud Computing, Load Balancing.

I. INTRODUCTION

The load balancing is important area of cloud computing environment. The process of balancing balanced the limited number of resource and unlimited number of job arrival. The process of balancing of workload improved the performance of cloud environment. The process of load balancing undergone in two different scenario static and dynamic. The static balancing technique used the process of CPU scheduling and some common data searching technique [13]. the dynamic load balancing adds some extra process and increase the cost of balancer. Now a day the dynamic load balancing technique used heuristic based function. The heuristic based function swarm intelligence for the purpose of balancing. The dynamic load balancing model consist of distributed computing technique for the controlling of load over the cloud environment. Now a day the public cloud computing environment used cloud partition technique. the cloud partition technique, divided the cloud load into serval part and then apply the process of balancing. In this paper used the graph theory based load balancing technique, these techniques divide the load in three situations one is under load, two is ideal situation and finally situation is overload. The all situated virtual machine mapped according to their total capacity of load. The all capacity of virtual machine sharing based [4]. The shared load by virtual machine map the situation of cloud environment. The haring of graph node imports the load of virtual machine for one node to another node. The time allocation frame slot used the concept of queuing theory for the handling the job according to the job frame [6]. GSO algorithm is especially useful for a simultaneous search of multiple optima, usually having different objective function values. To achieve this goal, a swarm must have an ability to split into disjoint groups. Otherwise, only one (local or global) optimum will be found. In GSO agents exchange information locally. Moreover, their movements are non-deterministic. In GSO, each glowworm distributes in the objective function definition space [9].

II. RELATED WORK

In this article [1] they described, Cloud Computing is a service that delivers computer nodes CNs resources, network equipments to a certain cloud user and shares the large scale of information, storage resources, computing resources, and knowledge for research. To gain high performance for the overall jobs' completion time and maximizing the throughput of cloud links, they discuss VMs placement that considers both computation resources and I/O data. The aim of this algorithm is to reduce the overall jobs' completion time. The Cloud-Sim Simulator results show that their algorithm can significantly maximize the overall application performance and reduce the average jobs' completion time compared among VMs placement approaches previously proposed in the literature.

In this paper [2] they discussed a scheme for distributed load balancing in the context of large data centers. They showed that duplicating jobs and sending replicas to different servers can significantly reduce the queuing time, even with a small number of replicas and in particular in high loads. They showed that, while their scheme is prone to degradation in presence of signal propagation delay, a simple heuristic may be used to overcome this problem. They described implementing the replication-based scheme in a real-world system is a simple task. However, determining the parameter configuration is highly dependent on the system condition, comprising the scale of the system, load pattern, job processing time, and inter-server delays. As different systems may be subject to different conditions, there is no single parameter configuration that is optimal to all systems.

In this paper author described [3] Cloud computing is the upcoming technology in current day scenario. It has emerged as a solution providing computing resources as a service to the consumers in the form of infrastructure, platform and software. This implementation aims at maximizing the resource utilization as well as profit for the service provider by cooperative game theory based approach for job scheduling in cloud environment. Additionally it also concentrates on minimizing the deadline violation and make span for the jobs submitted by the user. Thus, new job scheduling technique is proposed using the concepts of game theory and genetic algorithm.

In this work [4] Cloud settings provide excellent ser-vice isolation and packaging convenience. In many cases, cloud providers overprovision physical hardware to optimize for costs and real-world usage patterns. Modern hypervisors are able to handle CPU and memory contention caused by over-provisioning, but I/O contention poses a particularly difficult challenge due to the electromechanical nature of hard-disk drives (HDDs) that are frequently used in Big Data applications. The authors investigate how proactive disk scheduling, backed by predictive models and client-side coordination, can influence the overall throughput and responsiveness of a cluster in data-intensive computing environments.

In this implantation [5] they discuss a load balancing algorithm based on the method of estimating the end of service time in heterogeneous cloud computing environments. Scheduling cases of different levels were taken into account when they discuss formulas to calculate the average processing power of a virtual core. Simulation results showed that the proposed algorithm is more effective. The processing time and response time are improved in four scheduling cases. Especially, the cases of time-shared always give the best results. Load balancing directly affects the issue of datacenter power consumption with variety of workloads in the cloud. Load balancing helps effectively utilize computational resources, improve efficiency performance but it also causes the problem of energy consumption and carbon dioxide emissions.

In such a scenario [6] applications and data thereof can be hosted by various networked virtual machines (VMs). As applications, especially data-intensive applications, often need to communicate with data frequently, the network I/O performance would affect the overall application performance significantly. Therefore, placement of virtual machines which host an application and migration of these virtual machines while the unexpected network latency or congestion occurs is critical to achieve and maintain the application performance. To address these issues, this implementation proposes a virtual machine placement and migration approach to minimizing the data transfer time consumption.

In this paper [7] they enveloped a model of data center cooling for a realistic data center and cooling system. they simulated the model to obtain the cooling power as a function of data center load and outside temperature. This allowed us to investigate the impact of cooling on total cost, and explore whether cooling-aware load distribution can lead to cost savings. They also studied transient cooling effects resulting from abrupt, large changes in data center loads. they concluded that pre-cooling is necessary to prevent overheating in these scenarios. their policies incorporate this pre-cooling. Finally, they have shown that intelligent placement and migration of load can indeed lead to significant cost savings. Further, all electricity-related costs must be considered to maximize and en-sure consistent cost savings.

III. PROPOSED METHOD

In this section discuss the load balancing model using time allocation and graph based technique. the graph based technique compute the all capacity of virtual machine for the allocation of time. The total capacity of machine dedicated to the corresponding machine for the processing of balancing. The process of balancing describes here.

- Find capacity loads of all VMs based on the three condition define in graph allocation job is under load.

If

- Loader is balanced.

Exit

- Create the decision node for allocation

If T > maximum capacity
Load balancer not working
Else

- Call allocation process.

Share all virtual machine capacity
Call decision factor:
Create node of VMs

$$supply\ of\ VMS = Maximumm\ Capacity - \frac{Load}{Capacity}$$

Demand of each machine in node is

$$demand\ of\ wi_j = \frac{Load}{Capacity} - Maximumm\ Capacity$$

While Ta ≠ and WIJ ≠
For s=1 to # (Ta) do
Sort all VMs
For each task T in VMs find machine such as

Ta→VM_d $\min(\sum T) \in VM_d$ and $Load_{VM_d} \leq Capacity_{VM_d}$

If (T is allocated time)

$T_u \rightarrow VMd \mid \min(\sum Th) \in VM_d$

$T_0 \rightarrow VMd \mid \min(\sum Th + \sum Tm) \in VM_d$

$T_i \rightarrow VMd \mid \min(\sum T) \in VM_d$

$T_l \rightarrow VMd \mid \min(\sum T) \in VM_d$

The controller of graph controls all load according to their three allocation process according to dedicated time for the termination of job.

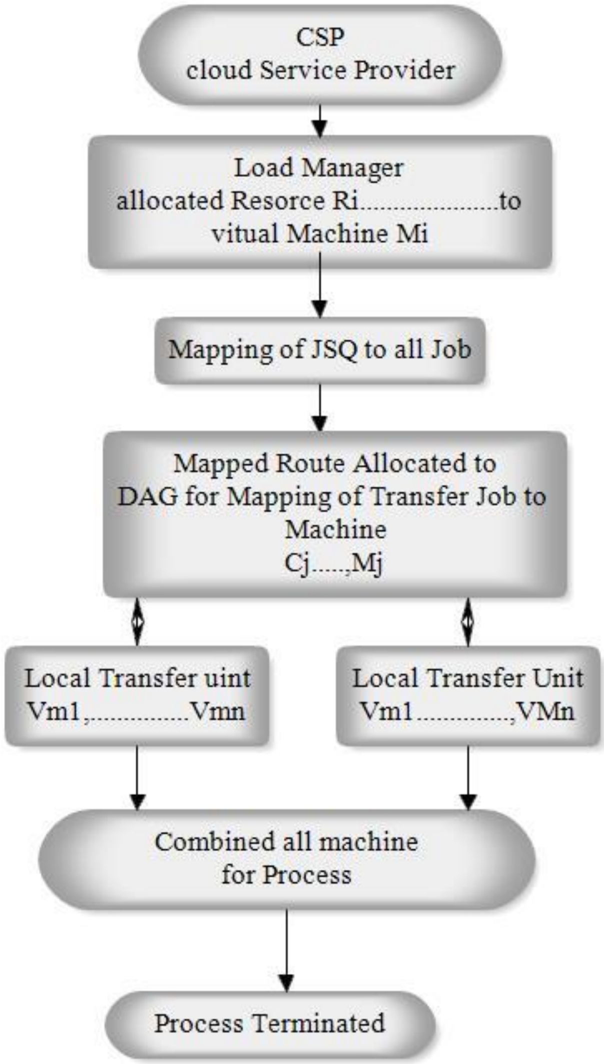

Figure 1: proposed model of cloud load balancing based on JSQ algorithm.

IV. EXPERIMENTAL RESULT AND ANALYSIS

To interact with various services in the cloud and to maintain the resources in a balanced manner to fulfill the requirement of resources/infrastructure by those services, several techniques are required. Based on a core set of features in the three common cloud services such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). To evaluate the performance of cloud computing techniques in cloud computing environment for the load balance and resource management, here we are using various numbers of techniques such as Round Robin, JSQ and IJSQ as a proposed method. For the further implementation and comparison for performance evaluation we used java programming languages with NetBeans IDE 8.0.1 tools for complete implementation/results process.

V. COMPARATIVE PERFORMANCE EVALUATION

Table 1: Shows the Response Time and Processing Time analysis for Round Robin Method.

Data Set	Method Name	Overall Response Time		Data Center Processing Time	
UB1 DC1	JSQ Method	Average	300.772	Average	0.366
		Minimum	241.639	Minimum	0.022
		Maximum	370.639	Maximum	0.638

Table 2: Shows the Response Time and Processing Time analysis for JSQ Method.

Data Set	Method Name	Overall Response Time		Data Center Processing Time	
UB1 DC1	IJSQ Method	Average	300.769	Average	0.365
		Minimum	241.639	Minimum	0.022
		Maximum	370.639	Maximum	0.638

Table 3: Shows the Response Time and Processing Time analysis for IJSQ Method.

Data Set	Method Name	Overall Response Time		Data Center Processing Time	
UB2 DC2	Round Robin Method	Average	701.881	Average	0.366
		Minimum	560.143	Minimum	0.019
		Maximum	875.135	Maximum	0.636

Table 4: Shows the Response Time and Processing Time analysis for Round Robin Method.

Data Set	Method Name	Overall Response Time		Data Center Processing Time	
UB1 DC1	Round Robin Method	Average	300.772	Average	0.366
		Minimum	241.639	Minimum	0.022
		Maximum	370.639	Maximum	0.638

Table 5: Shows the Response Time and Processing Time analysis for JSQ Method.

Data Set	Method Name	Overall Response Time		Data Center Processing Time	
UB2 DC2	JSQ Method	Average	701.881	Average	0.366
		Minimum	560.143	Minimum	0.019
		Maximum	875.135	Maximum	0.636

Table 6: Shows the Response Time and Processing Time analysis for IJSQ Method.

Data Set	Method Name	Overall Response Time		Data Center Processing Time	
UB2 DC2	IJSQ Method	Average	701.887	Average	0.367
		Minimum	560.143	Minimum	0.019
		Maximum	875.135	Maximum	0.636

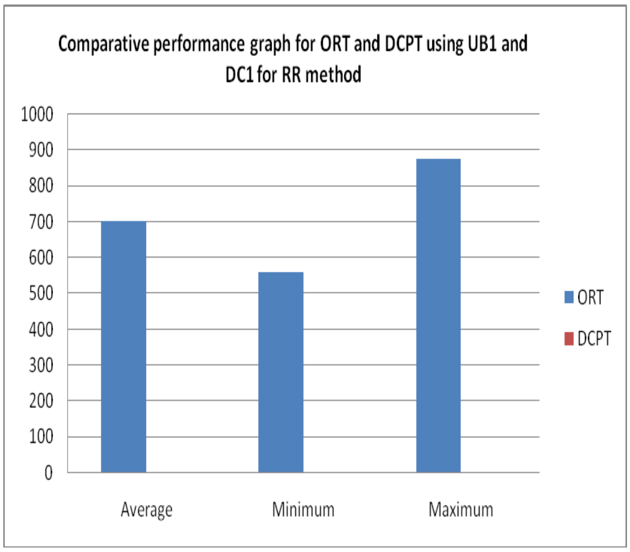

Figure 2: Shows the comparative performance of ORT and DCPT for UB1 and DC1 using Round Robin Method in terms of Average, Minimum and Maximum values in mili seconds.

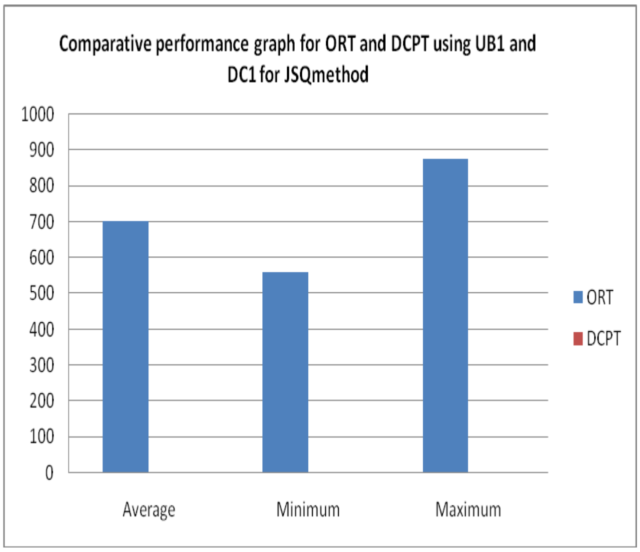

Figure 3: Shows the comparative performance of ORT and DCPT for UB1 and DC1 using JSQ Method in terms of Average, Minimum and Maximum values in mili-seconds.

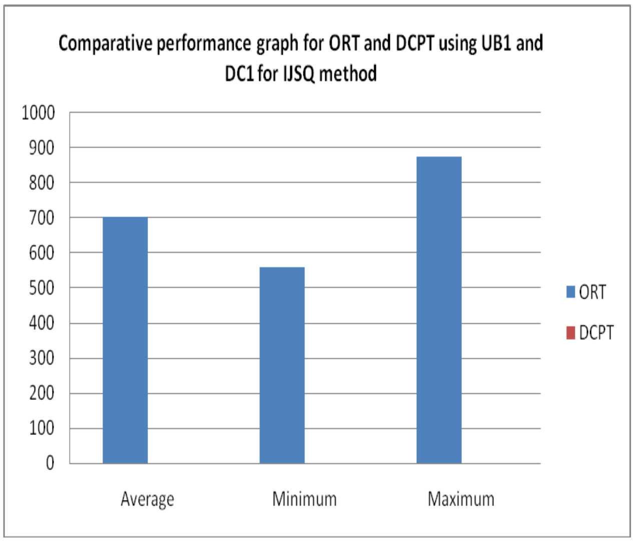

Figure 4: Shows that the comparative performance of ORT and DCPT for UB2 and DC2 using proposed IJSQ Method which shows the better processing and response time than methods such as round robin and JSQ in terms of Average, Minimum and Maximum time in mili seconds.

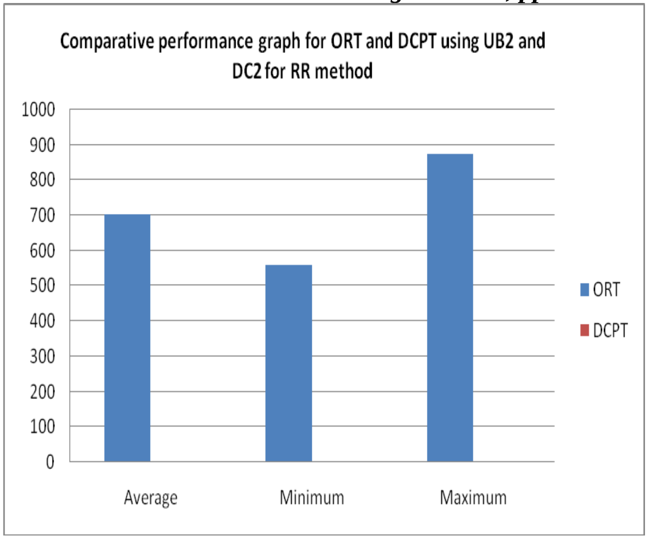

Figure 5: Shows the comparative performance of ORT and DCPT for UB2 and DC2 using Round Robin Method in terms of Average, Minimum and Maximum values in mili-seconds.

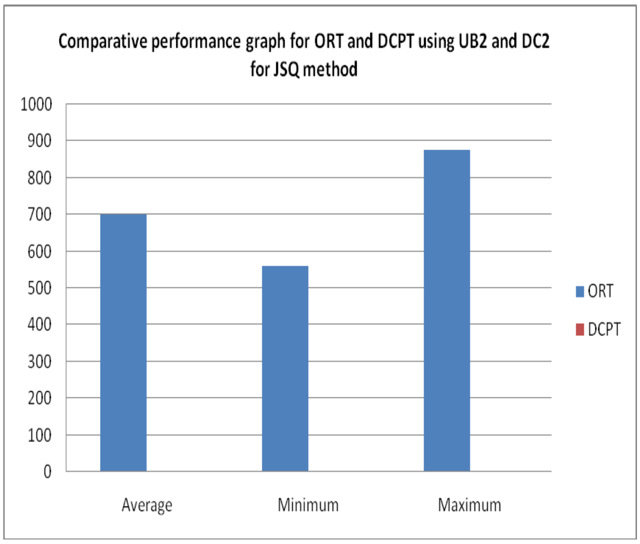

Figure 6: Shows the comparative performance of ORT and DCPT for UB2 and DC2 using JSQ Method in terms of Average, Minimum and Maximum values in mili seconds.

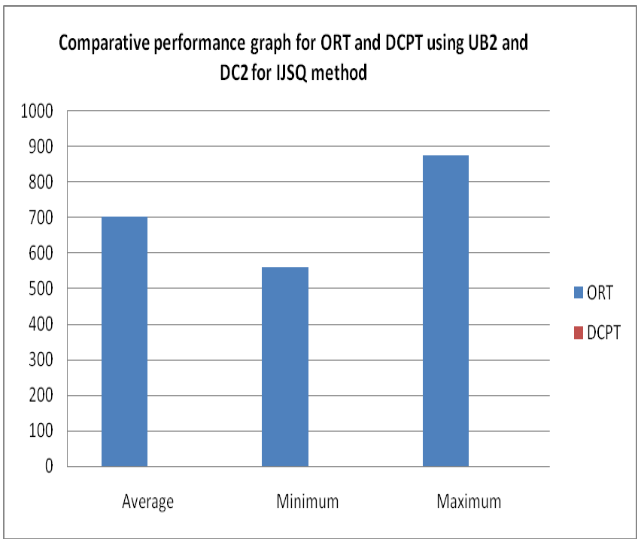

Figure 7: Shows that the comparative performance of ORT and DCPT for UB2 and DC2 using proposed IJSQ Method which shows the better processing and response time than methods such as round robin and JSQ in terms of Average, Minimum and Maximum time in mili-seconds.

VI. CONCLUSION AND FUTURE WORK

The cloud based services interact with user and dedicated cloud infrastructure. The interaction of user and cloud operation request to transfer the load to virtual machine and other resources. For the improvement of load efficiency in cloud computing various researcher and cloud designer used swarm based job and task scheduling technique. The swarm based task scheduling technique is very efficient in compassion of old and traditional technique such and FCFS and round robin technique.In this dissertation used JSQ algorithm and DAG allocation for load balancing policy in cloud environments. The IJSQ optimization set the diverse property of virtual machine and request job. The define fitness constraints function partially allocated job for dedicate machine and the distribution of job according to the process job scheduler.

REFERENCES

[1] Yasser Alharbi and Kun Yang "Optimizing jobs' completion time in cloud systems during Virtual Machine Placement", International Conference on Big Data and Smart City, 2016, Pp 1-6.

[2] Amir Nahir, Ariel Orda and Danny Raz "Replication-Based Load Balancing", IEEE, 2016, Pp 494-507.

[3] Alaka Ananth and K. Chandrasekaran "Cooperative Game Theoretic Approach for Job Scheduling in Cloud Computing", Computing and Network Communications, 2015, Pp 147-156.

[4] Matthew Malensek, Sangmi Pallickara, and Shrideep Pallickara "Minerva: Proactive Disk Scheduling for QoS in Multitier, Multitenant Cloud Environments, IEEE, 2016, Pp 19-27.

[5] Nguyen Khac Chien, Nguyen Hong Son and Ho Dac Loc "Load Balancing Algorithm Based on Estimating Finish Time of Services in Cloud Computing, ICACT, 2016, Pp 228-233.

[6] Jing Tai Piao and Jun Yan "A Network-aware Virtual Machine Placement and Migration Approach in Cloud Computing", IEEE, 2010, Pp 87-92.

[7] Kien Le, Jingru Zhang, Jiandong Meng, Ricardo Bianchini, Yogesh Jaluria and Thu D. Nguyen "Reducing Electricity Cost Through Virtual Machine Placement in High Performance Computing Clouds", ACM, 2011, Pp 1-12.

[8] Kyong Hoon Kim, Anton Beloglazov and Rajkumar Buyya "Power-Aware Provisioning of Virtual Machines for Real-Time Cloud Services", John Wiley & Sons, Ltd., 2011, Pp 1-19.

[9] Saurabh Kumar Garg, Adel Nadjaran Toosi, Srinivasa K. Gopalaiyengar and Rajkumar Buyya "SLA-based virtual machine management for heterogeneous workloads in a cloud datacenter", Journal of Network and Computer Applications, 2014, Pp 108-119.

[10] George Kousiouris, Tommaso Cucinotta and Theodora Varvarigou "The effects of scheduling, workload type and consolidation scenarios on virtual machine performance and their prediction through optimized artificial neural networks", The Journal of Systems and Software, 2011, Pp 1270-1291.

[11] Deepal Jayasinghe, Calton Pu, Tamar Eilam, Malgorzata Steinder, Ian Whalley and Ed Snible "Improving Performance and Availability of Services Hosted on IaaS Clouds with Structural Constraint-aware Virtual Machine Placement", IEEE, 2011, Pp 72-79.

[12] Sriram Kailasam, Nathan Gnanasambandam, Janakiram Dharanipragada and Naveen Sharma "Optimizing Service Level Agreements for Autonomic Cloud Bursting Schedulers", Parallel Processing Workshops, 2010, Pp 285-294.

[13] Zeratul Izzah Mohd Yusoh and Maolin Tang "Composite SaaS Placement and Resource Optimization in Cloud Computing using Evolutionary Algorithms", IEEE, 2012, Pp 590-597.

[14] Abhishek Gupta, Laxmikant V. Kale, Dejan Milojicic, Paolo Faraboschi and Susanne M. Balle "HPC-Aware VM Placement in Infrastructure Clouds", IEEE, 2013, Pp 11-20.

[15] Yue Gao, Yanzhi Wang, Sandeep K. Gupta and Massoud Pedram "An Energy and Deadline Aware Resource Provisioning, Scheduling and Optimization Framework for Cloud Systems", IEEE, 2013, Pp 1-10.

[16] Anton Beloglazov and Rajkumar Buyya "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers", John Wiley & Sons, Ltd., 2011, Pp 1-24.

