ISSN No:-2456 –2165

# Design and Theoretical Analysis of Virtual Machine Allocation based on time Slot Characteristics

Ranjitha R, (M.E.), P.Krishnamoorthy, M.Tech, Assistant Professor Computer Science and Engineering,
Kingston Engineering College, Katpadi, Vellore (D), Tamil Nadu, India

**Abstract — Cloud computing allows commercial customers Go up and down the use of resources according to your needs. Many of the gains promoted in the cloud model come from multiplexing resource through virtualization technology. We present a System that uses virtualization technology for dynamically allocate data center resources based on applications require and support green computing optimization of the number of servers in use. We present the "Time Slot Filtering" concept to measure the in equality in the use of multidimensional resources of a server. From minimizing the time, we can combine different types of workloads very well and improve the overall use of server resources. We develop a series of heuristics that to avoid effectively overload the system by saving energy used. Trial controlled experimentation and simulation results show that our algorithm achieves a good performance.**

**Keywords:** Data center in the cloud, Greedy heuristic, Automatic learning, Hyper-realistic, Linear integer programming, trade-off of energy network traffic, QoS, Evaluation of scalability, Time Slot Filtering.

## I. INTRODUCTION

The consolidation of the virtual machine (VM) [4] is widely used to minimize energy consumption based on the peak, outside the peak or the average CPU utilization of virtual machines [5] to compress them together in the minimum number of servers without degrading quality of service (QoS).However, the cloud workloads impacts dynamic nature of Consolidation in two aspects:

i)The CPU load correlation across VMs (i.e., the similarity of CPU utilization traces and the coincidence of their peaks).

ii) The data exchange across VMs (i.e., data correlation). In this context, several works use heuristics to either address CPU-load correlation, consolidating VMs when their peak utilizations do not coincide, or take data correlation into account. Nonetheless, jointly incorporating both metrics in a multi-objective optimization is an important aspect missing from prior works, as studied in a recent survey, which significantly increases the complexity of VM allocation. As the complexity rise, Integer linear programming (ILP) - based methods become unfeasible at runtime to provide an optimal solution. Similarly, heuristics are problem-specific and less sensitive to dynamic environments, and their benefits become limited for large problems. Thus, when tackling

dynamic problems with large state and/oration spaces machine learning (ML) methods, and in particular reinforcement learning (RL), are .However, in real DC scenarios, VM allocation faces the need to in,(QoS, network, etc.). This challenges the deployment of ML methods due to their limited Configurability. The proposal of balance methods that the trade-offs across these metrics, or dynamically change the optimization goals during run-time to meet Constraints, remains an open challenge, as it requires a Deep assessment on the previous techniques, together with The integration of their strengths. Within this context, hyper-heuristics are a promising solution to leverage the benefits of VM allocation approaches. Hyper-heuristicsareheuristics that choose heuristics and allow determine in which method to use depending on the current DC status, providing better trade-offs than when using the methods separately.

### *A. Disadvantages*

- Requests are more than cannot access server.
- Server response slows down. User consumption time

## II. PROPOSED SYSTEM

First we propose and evaluate two different ones Approach has to address the problem of VM allocation:

- A two-phase greedy heuristic.
- A ML-based approach Both approaches explode CPU-load and data correlations,

Together with information on the topology of the DC network. Our strategies consolidate virtual machines in the minimum number of server and rack and set dynamic voltage and frequency resized (DVFS) appropriately. So, let's introduce a novel hyper-technical method that integrates the strengths of both heuristic and ML methods. We evaluate our approaches in terms of energy consumption, degradation of QoS, network traffic, number of migrations and scalability and comparison to an optimal solution based on ILP and state of the art methods. In particular, our main contributions are follows: We propose a hyper-objective multi-objective lens method to dynamically determine which method to insert Heuristic and ML must be used at all times. to. We offer two virtual machines with energy and network recognition allocation methods:

- A greedy two-phase heuristic
- A low complexity ML based approach that uses value iteration algorithm to assign virtual machines to servers. Second. We provide an assessment of flexibility, scalability, advantages and disadvantages of heuristic vs. ML for the extremely dynamic and complex VM allocation problem. do. We compare our proposed solutions with a ILP based method, which provides an optimal solution, even with two state-of-the-art methods.

### *A. Advantages*

- The server will have a rush hour can access more application.
- If the server has no time limits, you will have access minimum request.
- It is possible to analyze and set the time. limit to the server.
- Avoid wasting time.
- The server response will be fast.

### *B. Modules*

1. Cloud computing.

2. Supply of resources in the cloud.

3. Positioning of the virtual machine.

4. Quality of services.

5. Filtering time intervals.

### 1. Cloud computing

- Add service details and customer service websites.
- Manage data center memory, CPU usage.
- Setting up and creating computers environment.

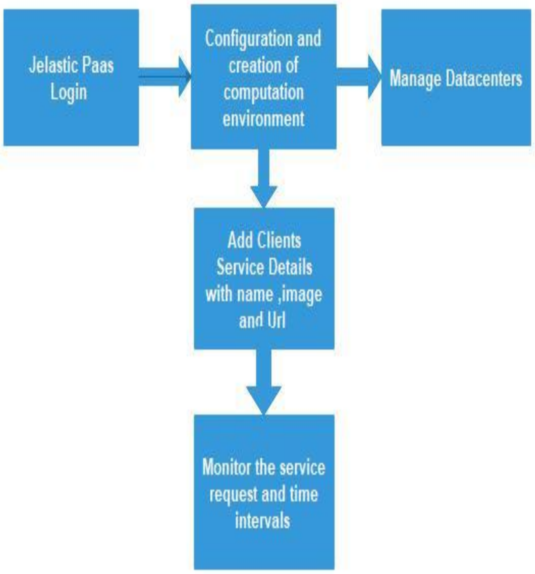

Jelastic Paas Login
Configuration and creation of computation environment
Manage Datacenters
Add Clients Service Details with name ,image and Url
Monitor the service request and time intervals

Fig 1: Cloud computing

**Input**: customer service details, customer service URL.

**Output**: successful addition of service details, administration environment.

### 2. Provisioning resources in the cloud:

Provisioning resources in the cloud are challenging tasks that can being compromised due to the lack of availability of what was fore seen resources. Quality of service (Qos) requirements of workloads derives from provisioning means to cloud workloads.

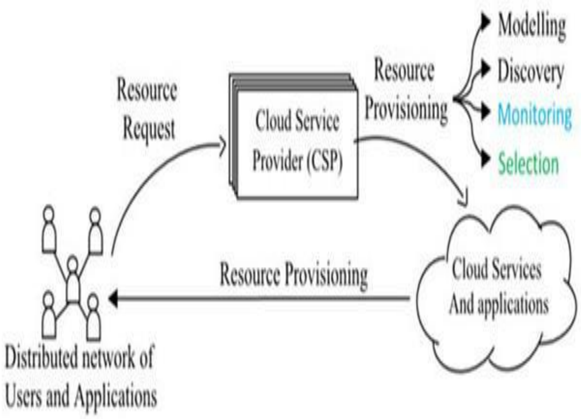

Resource Request
Cloud Service Provider (CSP)
Resource Provisioning
Modelling
Discovery
Monitoring
Selection
Cloud Services And applications
Resource Provisioning
Distributed network of Users and Applications

Fig 2: provisioning of resources in the cloud

Discovery of the best couple of workload resources based on cloud application application requirements are an optimization problem. Failed to provide acceptable Qos to the cloud users until the supply of resources is offered as a crucial element Capacity. Provisioning of resources based on Qos parameters therefore, a technique for efficient provisioning is required resources.

### 3. Location of the virtual machine:

When a virtual machine is deployed on a host, the process to select the most suitable host for the virtual machine known as virtual machine placement or simply positioning. During placement, hosts are classified according to virtual hardware and resources requirements of the machine and the early use of resources. Host ratings also take into consideration of the employment Objective maximization in individual hosts or load balancing between Guests. The administrator selects a host for the virtual machine based on guest ratings. The position of the virtual machine is the mapping process from virtual machines to physical machines machinery.

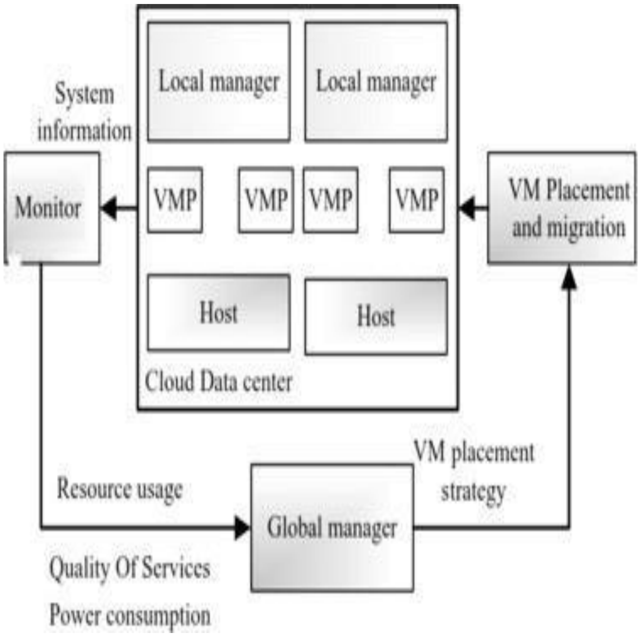


Fig 3: position of the virtual machine

In other words, the positioning of the virtual machine is the process of selecting the most suitable host for the virtual machine. the process involves the categorization of virtual machines hardware and resource requirements and the intended use of resources and the goal of positioning. The goal of positioning can or maximize the use of available resources it can save energy by being able to close some of them server. The positioning of the autonomous virtual machine the algorithms are designed taking into account the previous objectives.

### 4. Quality of services:

QoS (Quality of service) refers to a large collection of network technologies and technologies. The goal of QoS this is provide guarantees on the ability of a network to provide predictable results Elements of network performance within the scope of the QoS often includes availability (time of activity),bandwidth (performance), latency (delay) and frequency of errors. QoS involves prioritization of network traffic. QoS can be directed to a network interface, a server or the performance of the router or in terms of specific applications. There The network monitoring system should generally be implemented as part of QoS, to ensure that networks work in the system.

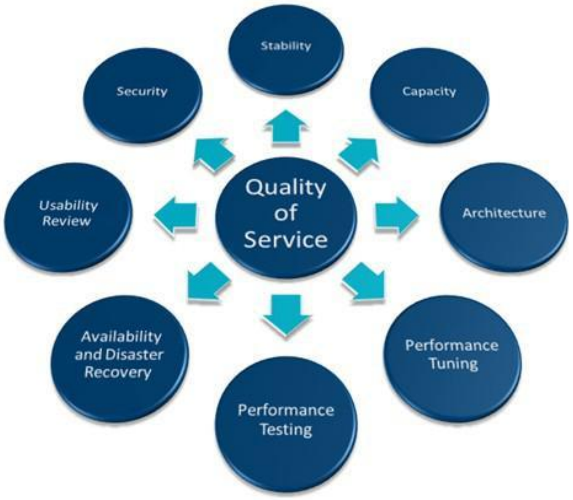


Fig 4: quality of services (QoS)

The desired level QoS is particularly important for the new one generation of Internet applications such as VoIP, video request and other consumer services. Some main networks. Technologies like Ethernet have not been designed to support priority traffic or guaranteed performance levels, production QoS solutions are much more difficult to implement Internet.

### 5. Time interval filter:

For the activities of the instance, make workflow much more efficient programming. Filtering the time interval though there are many spaces available, not all have been found activity requirements of workflow instances. A bit 'the available time intervals may not be available for an activity even before the service is assigned.

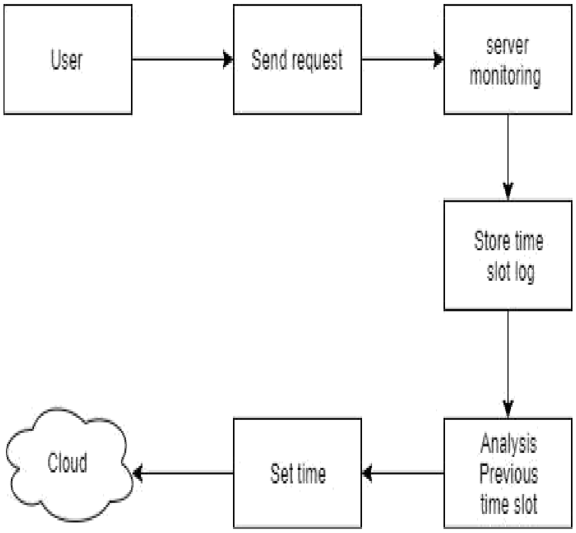


**Fig. 5: filtering the time interval**

For example, Eft (n)> D in the fastest program or the duration of a time interval is less than the execution time of activity or start or end time is beyond the first beginning or the last moment of completion of the activity. When everything is filtered Impossible time positions, the remaining time slots are eligible.

## III. PROPOSED ALGORITHMS

The ILP method can be divided into two minimizations problems:

Energy consumption (ILP-Power), and ii) data communication (ILP data). Because of the variable nature ofDC workloads (that is, the virtual machine usage patterns change now), there is an optimal VM allocation for each time interval and, therefore, the need to invoke the ILP-based method. The weather The slot life is a parameter that can be adjusted by the DC operator based on the granularity of the traces used, for increase accuracy We define ILP formulations in a generic form regardless of the network topology Therefore, an optimal server for the data server you get the communication that represents the total network traffic. In the following subsections, we describe These two optimization goals in detail.

- *Proposal For Bifurcation Of The Two-Phase Effect Method*

In this section, we propose a greedy two-phase heuristic Algorithm (heuristic, as follows), at any time T slot, minimizes the power consumption -PT s etotal -TD network traffic (Phase 1), then assign traffic resulting in a network aware fashion network(Phase 2)

A. Phase 1: VM grouping

We divide this phase into two phases and use a method similar to that presented in [6]. First of all, in the time interval T, all the virtual machines available in the system are represented as points in a two-dimensional plane (2D). Depending on data and CPU load correlation properties, such as virtual machines highly correlated with data it should be grouped while the CPU load is relatedVMmust be separated, a function is defined to calculate forces of attraction and repulsion between each two virtual machines. However, unlike the original algorithm, calculate the attraction force as the worst case spike bidirectional data exchanged between two virtual machinesduringtime interval Similarly, the repulsion force is calculated as worst CPU usage peak when the peaks of two VM matches during the last time interval. As a result, the points are reassigned in the 2D plane with new coordinates based on the calculated forces.

B. Phase 2: assignment of conglomerates

At this stage, we assign the conglomerates to the appropriate groups Servers considering the DC network structure as described in algorithm 1. This algorithm fills the frames one by one, Reduce the number of active switches and minimize Network power, maintaining excellent communication Servers close to each other.

*Algorithm 1: Assignment of groups*

Input: network topology and data

Communication graph = {V, E}

V = cluster and W (E) = data communication

Among the clusters.

Output: group assignment

1: Edgemax <---- Max. Edge (W (EW; z) + W (Ez; w))Between any Vw and Vz

2: NTh agr <----- 0 Initial network traffic of hth agr. Change

3: for r = 1: total rack

```
4: NTr tor <---- 0 Initial ToR switches network traffic of rth rack.

5: unused rth rack servers <---- Total rst rack servers

6: while (Selected clusters <= unused servers in the rst rack) &(NTr tor <= Btor) and (NTh Agr <= Bagr) do

7: If Vw and Vz were not assigned, then

8: Assign the w and z clusters to two servers in the rst rack

9: NTr Tor <------ Updates rack server traffic

10: NTh Agr <------ Update the traffic of the racks of the hth group

11: Update unused servers in the rst rack

12: if Vw or Vz were not assigned then

13: Assign w or z to a server in the rst rack

14: NTr tor <---- Rack server traffic update

15: NTh agr <------ Update the traffic of the hth racks group
16: Update unused servers in the fifth rack

17: final if

18: Combine Vw and Vz

19: Update W (E)

20: if all the clusters are assigned then

21: Fine

22: final if

23: Edgemax <------- Find the maximum weight of the edge
24. Find the number of selected clusters ('1' or '2') when
both
the groups were not assigned

24: final while

25: end a
```

**A**. Optimization of energy consumption

The proposed task of ILP-Power VM is aimed at minimize the overall power of the server. The goal of minimization is given by the eq., where PT e PT j; n indicates the power consumption of the global and j-th server the umpteenth sample of the time interval Tth, respectively. Ns and New Testament are the number of servers in the DC and the number of samples in a time interval, respectively. The binary variable XT j is defined to indicate if the jth server is activated (XT j = 1) Off (XT j = 0) in the time interval Tth. We use the track variable PlaceTj; k to indicate whether kth VM (k 21; 2; :::;NVM) is located on the jth server in the Tth tme slot. NVM is the Total number of virtual machines available in the DC. Matrices VMcpuTk; n and memTk VM; n, contains the k-th VM Memory memory and memory usage in the umpteenth sample, respectively during the time interval Tth. Likewise, memj UT;n indicates the total use of jth server memory. Restriction 1 requires that each virtual machine be placed only in one server.

**B. Optimization of data communication**

The amount of data exchanged directly between virtual machines affects network traffic and response times. In practice, twoVirtual machines regularly exchange a variable amount of data. Our purpose is to minimize total data communication (network traffic, Total) between the servers. In our formulation, DTj; north represents the data communication of the Jth server; that is, the amount of data transferred from a server in the nth sample of the time interval Tth. To express DTj; n, the binary variable BinVM status Tj; k; the; k; l indicates whether both adults The virtual machines have been assigned to the jth server (Bin VMstatusTj;k;l; kth; k; l = 0); otherwise, Bin VMstatusTj;k;l; k; l = 1 in the tenthtimespace. The VMdataTk array; In contains the amount of data transfer lost from kth to lm VM nell' ennesima Time sampling Tth Slot.

**Algorithm 2: hyper-historical algorithm**

```
Input: Oi T-1 = {PDC T-1, WCVT-1, TNtor T-1, TNagrT1;TNcr T 1}, i <--- M

Output from the hash table: select a method of M

1: Hash <------ Hash Generator (OMT-1)

2: hash observed <----- Is Hash Observed (hash, Hash table)

3: If Hash observed == True, then

 4: m <- Select method with min (Cost Hash = Numb Hash), I'm

5: more if Hash observed == False, then 6: hash register in the hash table.

6: m <----- Select the method with a minimum cost Function (OT1)

7: final if
```

8: Executes the m method for the time interval T

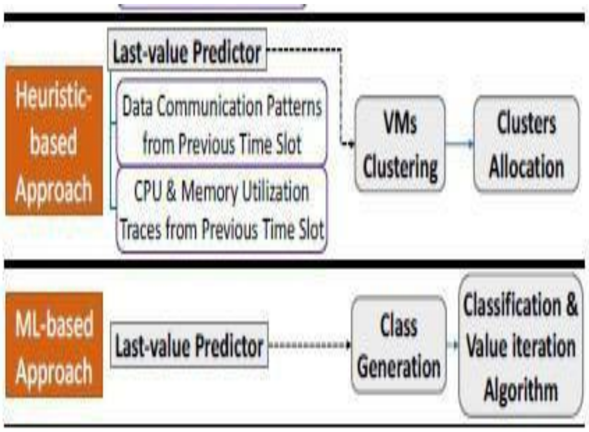
9: get Oi T At the end of the time, slot T;I

10: Cost i <---- Function cost (Oi T), <----- i

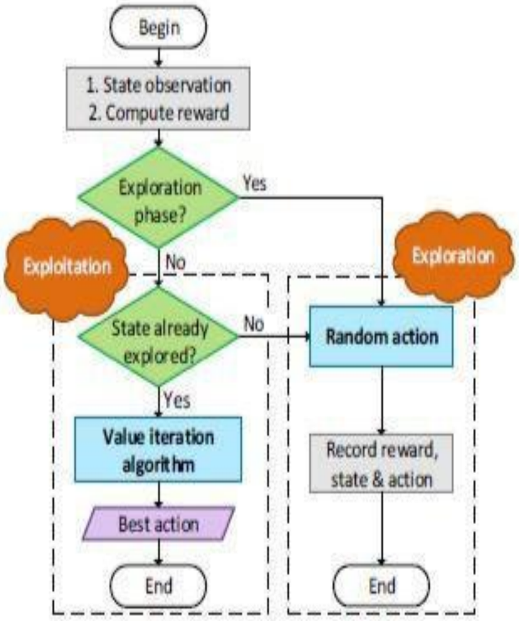
11: m <----- Search method with min (cost)

12: Cost m Hash <----cost m Hash + Cost m

Num m Hash <------- Num m Hash +1 For the last three components, present three initial solution construction strategies, two methods of improvement and one disturbing strategy.


Heuristic-based Approach
Last-value Predictor
Data Communication Patterns from Previous Time Slot
CPU & Memory Utilization Traces from Previous Time Slot
VMs Clustering
Clusters Allocation
ML-based Approach
Last-value Predictor
Class Generation
Classification & Value iteration Algorithm

Algorithm 1: Cluster Allocation



Algorithm 2: Hyper-heuristic Algorithm

## IV. CONCLUSIONS

For the last three components, we present three initials strategies for building solutions, two methods of improvement and a breaking strategy. energy, network traffic, QoS, migration and scalability. We presented the design, implementation and evaluation of a resource management system for cloud computing services. Our system multiplexes virtual resources to physical resources adaptively depending on the change. request. We use the asymmetry metric to combine virtual machines with different resource characteristics appropriately, so that the server's capabilities are used correctly. Our algorithm achieves both overload prevention and green computing for systems with multiple resource constraints.

Many efforts have been made to reduce energy consumption in data centers. Hardware-based approaches include a new thermal design for less cooling power or the adoption of low-power and power-efficient hardware. The work uses dynamic dynamism of voltage and frequency (DVFS) to adjust CPU power based on its load. We do not use the DVFS force calculation, as explained in Section 7 of the supplemental file. Powernap uses new hardware technologies, such as the solid state drive (SSD) and Self-Refresh DRAM to implement a fast transition (less than 1 ms) between the entire operation and the low power state, so you can "take a nap. ""in short and inactive intervals.

## V. FUTURE WORK

In the future we can add additional performance categories to reduce the flow of energy and measure the energy saved and the energy lost.

Since this work is only a conceptual structure, more work is needed to implement the structure and solve new problems. Some important points are:

### *A. Rules of the division in the cloud*

Division into the cloud is not a simple problem. Therefore, the framework will require a detailed divisional methodology in the cloud. For example, the nodes of a cluster could be far from the other nodes or there will be some clusters in the same geographic area of the theater that are still very far apart. The division rule should be based simply on the geographical position (province or state).

### ***B. How to configure the update period***

In the analysis of statistical data, the controller and the cloud sections must regulate the information for a certain period of time. If the period is too small, the higher circuit will affect system performance. In the long run, the information will be too old to make a good decision. Therefore, the dimensions and data tools are necessary to correctly set up updates.

### *C. A better evaluation of the charge status*

A good algorithm is needed to configure the high load level and the low load level, and the evaluation mechanism must be more complete.

### *D. Find another load-balancing strategy*

Other load-balancing strategies can provide better results, so tests are needed to compare different strategies. Many tests are needed to ensure inefficiency of system availability.

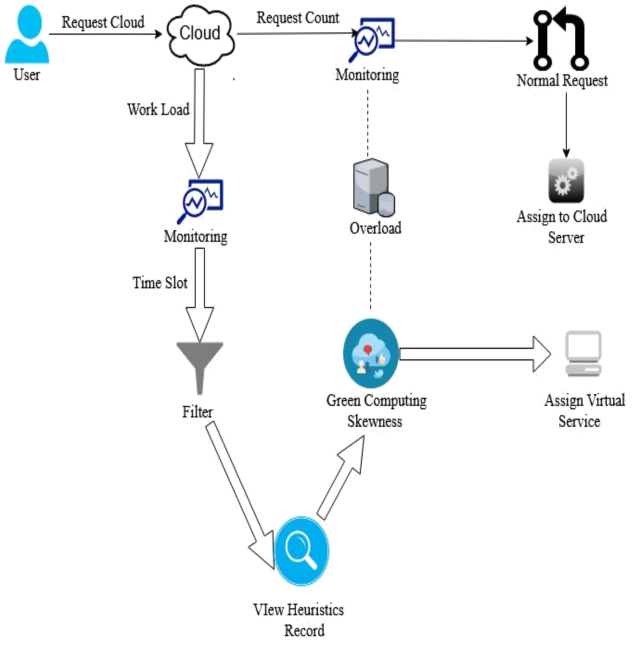


Fig 6. Architecture Diagram

## REFERENCES

[1] M. Zapata et al., - Frequency-responsive refrigeration management to improve server energy efficiency, IEEE transactions in parallel and distributed systems (TPDS), vol. 26, no. 10, pp. 2764-2777, 2015.

[2] L. A. Barroso, J. Clidaras and U. Höolzle, -The data center as a computer: an introduction to the design of warehouse-scale machines, second edition, Synth Lect Comput Archit, vol. 8 (3), pp. 1-154, 2013. [2] X. Meng, V. Pappas and L. Zhang, -Improve the scalability of data center networks with the location of virtual machines with traffic recognition, in IEEE Conference on Information Communications (INFOCOM), 2010, pp. 1154-1162.

[3] E. Pakbaznia and M. Pedram, "Minimizing Cooling Costs and Server Power Costs" at the ACM / IEEE International Symposium on Electronics and Low-Power Design (ISLPED), 2009, pp. 145-150.

[4] A. Beloglazov and R. Buyya, "Managing Overloaded hosts for dynamic consolidation of virtual machines in cloud data centers based on service quality restrictions, IEEE TPDS, vol.24, n.7, pp 1366-1379, 2013.

[5] M. Zapata et al., Leak detection refrigeration management to improve server energy efficiency, IEEE transactions in parallel and distributed systems (TPDS), vol. 26, no. 10, pp. 2764-2777, 2015.

[6] LA Barroso, J. Clidaras and U. Höolzle, "The data center as a computer: an introduction to the design of machines on a warehouse scale, second edition, Synth Lect Comput Archit, vol 8 (3), pp 1 -154, 2013

[7] X. Meng, V. Pappas and L. Zhang, - Improve the scalability of data center networks with the location of virtual machines with traffic recognition, in IEEE Conference on Information Communications (INFOCOM), 2010, p. . 1154-116

[8] E. Pakbaznia and M. Pedram, ―Minimizing data center cooling and server power costs, in ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED), 2009, pp. 145–150.

[9] A. Beloglazov and R. Buyya, ―Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints, IEEE TPDS, vol. 24, no. 7, pp. 1366–1379, 2013.

[10] Z. A. Mann, ―Allocation of virtual machines in cloud data centers–a survey of problem models and optimization algorithms, ACM Compute. Surv, vol. 48, no. 1, pp. 11:1–11:34, 2015.

[11] A. Iran far, S. N. Shahsavani, M. Kamal, and A. Afzali-Kusha, ―A heuristic machine learning-based algorithm for power and thermal management of heterogeneous mpsocs, in ISLPED, 2015, pp. 291– 296.

[12] P. Cowling, G. Kendall, and E. Soubeiga, ―A hyperheuristic approach to scheduling a sales summit, in International Conference on Practice and Theory of Automated Timetabling III, 2001, pp. 176– 190.

[13] S. Esfandiarpoor, A. Pahlavan, and M. Goudarzi, ―Structure-aware online virtual machine consolidation for datacenter energy i improvement in cloud computing, Computers & Electrical Eng., pp. 74–89, 2015.

[14] D. Meisner et al., ―Power management of online data-intensive services, in ACM Int. Symp. on Comput. Archit. (ISCA), 2011, pp. 319– 330.

[15] Suraj Pandey, LinlinWu, Sideward Mayra Guru, Rajkumar Buy. "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing.

[16] Fatimah Farahnakian, Pasi Liljeberg, and Juha Plosila. "Energy-EfficientVirtual Machines Consolidatio in Clou Data CentersusingReinforcement Learning"(2014).

[17] Ts`epoMofolo, R Suchithra. "Heuristic Based Resource Allocation Using Virtual Machin Migration A Cloud Computing Perspective".(2013).

[18] Xiaoqiao Meng, Vasileios Pappas, Li Zhang "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement".

[19] Kejiang Ye, Dewey Huang, Xiaoyong Jiang, Haunt Chen, Shang Wu."Virtual l Machine Based Energy-Efficient Data Center Architecture for Cloud Computing A Performance Perspective".(2010).