International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 3, Issue 5, September-October 2014 **ISSN 2278-6856**

Thrusting Energy Efficiency for Data center in Cloud Computing Using Resource Allocation Techniques

Vijaya -Kumar-C [1], Dr. G.A. Ramachandra[2]

[1] Computer Science and Technology, Sri Krishnadevaraya University
Anantapuramu, Andhra Pradesh, India

2 Computer Science and Technology, Sri Krishnadevaraya University
Anantapuramu, Andhra Pradesh, India

Abstract

In recent development, complexity data intensive applications are increasing in cloud computing. We know that it can reduce investments, human resources and enhance productivity. Data centers play a key role with rapid growth online services of client demands in terms of providing the infrastructures as services (Iaas). For data intensive application requires more number of data centers and also massive amount of energy used to operating the servers. Due to increases in data centers in different locations its impact on environment in terms of increased the carbon footprint. We proposed virtual machine migration(VMM) technique to optimize data centers, satisfy performance resource allocation, reduce the server failures and also energy consumption. To reduce the energy consumption, we are proposed virtual machine placement and dynamic load balancing algorithms.

Keywords: Data intensive applications, Data centers, VM migration, VM placement, Dynamic load balancing.

1. INTRODUCTION

Many people are adopting the cloud computing due to their priority of providers and customers in terms access speed, transfer rate and storage capacity in servers which impact on the environment. Due to increase pollution it creates to increase the temperature, floods and droughts. To reduce the pollution in the atmosphere, it is necessary to reduce natural resources in which generate the electricity and it play major role to increase the co2 in the atmosphere[1]. For highly electric power consumptions by hardware such as servers, monitors and cooling systems that will impact on the environment. In this context, cloud computing an emerging model for represented the data intensive applications and highly design infrastructures and power consuming data centers to supporting the elasticity, scalability and increase the performance of user requirements. So that we concern the energy consumption in the cloud computing, migration of services to access the remote server increases network traffic and it will impact on the data centers. We mentioned in two levels that is data centers level and network level. In data centers level, we consider power consumptions in servers, power supply, data storage, cooling infrastructures and maintain the all compute resources and in network level, we choose to maintain the virtual machines, core network level and dynamic load balancing[4]. Virtualization influence in the IT sector and increase performance of energy efficiency in data centers, it can't reduce energy consumption but we can optimize data center infrastructures to support for dynamic high density computing environment. To maintain the data center is very expensive and intensive process. In cases the data center is single point failure, there is possible to damage from fire, earthquakes and others environment issues, to avoid failures we create duplicate data center and that are access from the remote location.

2. SYSTEM MODEL

Energy consumption model explains relation between CPU utilization in data centers. Based on the CPU utilization, we can identified how many VM resources available for service to allocate users[18]. We approach to handling energy-aware scheduling in data center which provides an allocations VM, reduces the number of activate nodes and switch off the number of deactivated nodes using the workload consolidations.

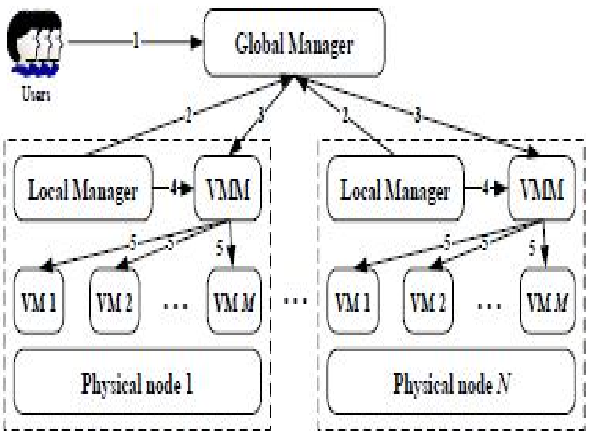


Figure 1 : SYSTEM MODEL

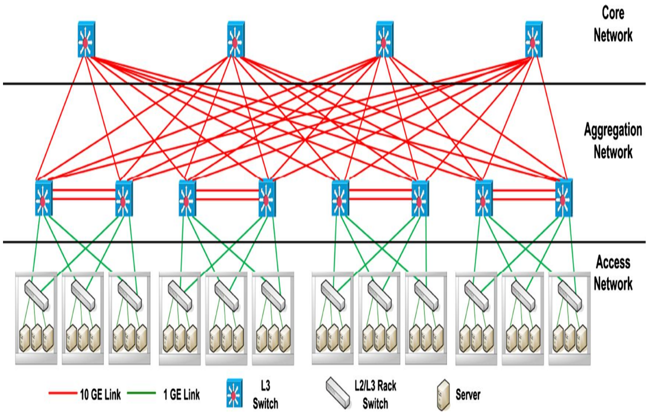


Figure 2 Architecture of Cloud Data Center

Several research works are implementing advancements in hardware technologies [41] such as efficient computer monitors, low power consumptions CPU and efficiency drivers having high quality. Virtualization is the one of the most effective steps to improve the energy efficiency in data centers. In this model we consider task consolidation problem in virtual machines and the resources are incorporated with effective power consumption for idle state or inactive mode virtual machines In task consolidation problem we assign the set of number of n tasks and m resources, the main intension of this problem is utilized the maximum resources and minimize the energy consumption in data centers[18]. We have identified the Static power management (SPM) and Dynamic power management machine (DPM), we proposed dynamic adaption to allocate the virtual machine at run time. So that we can estimate utilization of energy and implementing the live migration and make the changes idle nodes to sleep nodes or sleep nodes to idle nodes. From this we can minimize power consumption when the servers are in active mode. For given workload and power conditions, it first solve the problem using a simple optimization algorithm and find an optimal sequence of adaptation performance workload we proposed advanced self-aware A* search algorithm implemented and tracking a desire the utility layer levels and it use it to prune reduce the cost of energy consumption. In large scale distributed computing system, the power and energy management methodologies can be classified into two categories a). Static energy management techniques b) Dynamic energy management techniques.
Flow of Energy consumption: Chip→Platforms→Server→Racks→Clusters→Datacenter

3.RELATED WORK

Resource Allocation Scheduling techniques can be classified into (a) Network-aware resource scheduling (b) Thermal-aware resource scheduling (c) Virtualized resource scheduling. Virtualization and server consolidation works to reduce the number of activated servers, minimize hardware devices and labors. In servers, workloads reach peak level, we have to analysis the CPU utilizations and individual VM are allocate on physical hosts to minimize the active servers with maximum work load . We have used Live VM Migration concept and implemented into the consolidation to decrease the Virtual machines. Each Virtual machine can encapsulate multiple applications as similar to the Physical machine such as DNS, SMTP, Web server and remote desktop, etc.We used DRA workload consolidation framework to managed multiple Virtual machine from single Physical machine and dynamically arrange VM placement to increase the application performance and reduce the power consumption. Each node runs an manager on domain 0 and collects the statistics information of resource allocation for each VM on the node. By observing the CPU and network usage we can calculate the scheduling events in hypervisor, sometimes the memory usage more than the VM capacity. So all the applications in the VM are not responding within time slots, memory shortage of VM we have approach the swap activities. Unfortunately, the guest OS is required to install the swap partitions, it make memory allocations by the time swapping occurs. In collection of statistics, each PM are forwarded to the central controller schedule to the Virtual machines and Virtual machines scheduler receives history message in periodically from the manager .Based on the previous load history of PMs, dynamic resources are allocated in a Virtual machines and we predicts the future load in PMs and also VMs also predicts the resource demands of VMs. We can minimize the data by using the compression techniques and we classified the data into three types a). video format b). document ,text, file format c). database file format like oracle, sql, etc. We make as default priority data format Video, Database file format files and doc file format and we have also give the priority based on the user request or demand.

3.1 Implementation of Data reduces using Map Reduce.

Map Reduce is framework developed by Google that allows users easy to distribute the large intensive applications in different domains. In this model we work on to the map the data, reduce the data, and combine the data for user requests, partitions and also sorting the data. We used two main functions are Map function and Reduce function[6]. The Map function takes output as key and values pair and it will manage the distribution and execution of number of tasks and gather output and reports as per the user requests. It will start with input partition for their individual reduces tasks and reduce function based on the each unique input key by the users. The main responsible for distributing the number of jobs among the number Task tracker nodes of the cluster and it will collect the storage of output in N parts for N number of reduce tasks. The Hadoop framework provides the different types of input format and we have implemented KeyValueText and Sequence File input formats.

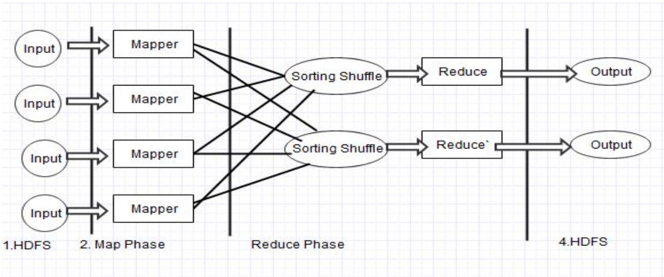


Figure 3 Map Reduce Workflow

HDFS: In this Phase data send to their distributed file system on different unstructured data, Semi structured and Structures. MapPhase: The input file to the Map Reduce file system and number of split the file into smaller files after files are converted into key and values format. Reduce Phase: After MapPhase, a network intensive jobs begins to send the intermediate key and values from MapPhase to Reduces Phase and sorting /shuffle operation is applied.

1. Generate each task in map based on the task duration distribution
2. Allocate the tasks on predefined number of jobs using Round Robin + priority algorithm
3. To find the estimated time and last reduce task job.
4. Repeat 1-3 for number of iteration to get the effective outputs.
5. To analyze the outputs from the jobs to get the expectations of users.

Map Reduce algorithm

Step1: Map Reduce framework input consists of a set of input file formats (file1, file2, file3....file N).

Step 2: If there are K sources and L receivers, then number of input files will be N = K*L.

Step 3: The Primary Key of the mappers is index of the Key and the value of the file.

Step 4: In the mappers each file is migrated by using sequential format and map function is migration algorithm working on file.

Step 5: The output file of a mappers is values for Data[1][1],Data[2][2],Data[3][3],............. , Data[M][R]. In few scenarios the elements may have zero values.

Step 6: A set of Data[i][j], 1≤ i ≤ M, 1≤ j≤ R is the output file of mappers and it will repeated from sources to each receivers .

Step 7: Sort the input files based on the Key and values and group stage. For each Data[i][j] all files are sorted and grouped.

Step 8: Sorted order data[i][j] indexes, Data [0][1],[1][1],[1][2],.....................[M][R], and the values of each element are the value of each occurrence of the element.

Step 9: Data [1][1] is repeated and make the values {value1, value2,value3}.

Step 10: In Reduce phase , all the occurrence of each values of data are formed as final output file format.

4. Dynamic Load Balancing Algorithm

In cloud computing provides an organization to get the effective ability and distribute application requests to the 'N' number of application deployments /Services in data centers. To increase the performance of server, it is ability to shift loads dynamically across the world through the internet and whenever excess traffic are sending request to server it will degrade the server utilization [5] . So to avoid cloud bursting, we are predictive load balance and shifting of load is automated across the global. We have configured with the global server to maintain the excess traffic. In a distributed system, dynamic load balancing adapting the distributed schema and non distributed schema. In distributed schema the dynamic load balance is executed by all nodes in the server and it will sharing load balancing for every user based on the priority of task. In few scenarios when the nodes are failures it will not effect to the remaining nodes. So that we can make load balance halt for failure nodes and sharing the load balance remaining active nodes in the server for improving the server performances. We are proposed Centralized dynamic loading algorithm to overcome the excess traffic, we make the load distribution based on the current work load and also predictive of feature workload.

Scenario1: In few cases consolidation workload of 'X' nodes are consumed more energy and ' Y' nodes are consumed less energy to make them as balanced and send request to the Global server and balanced the workload according to their priority. For example, consider five identical servers having same capacity of size, A,B,C, D and E to their relative loads are 90 %,70 %, 50 %, 30% and 10 % respectively. To make them as perfect load balancing all servers should maintain 50 % of each server's capacity. To minimizing the load balance for every server we have distributed the all servers into centralized and combine the servers. In our example, the load from E will migrate to A and load D will migrate to B and no need to migrate with other servers of C indivisible. For distribution of load balance and migrate the load to appropriate servers we can improve the QoS for access the data from nodes.

Scenario 2: In consolidation workload of 'X' nodes are consumed more energy and 'Y' nodes are consumed more energy, to make them as balanced to send the request to the Global server and balanced the workload and in cases of servers till having more excess traffic, create new servers as per their priority. For example, consider five identical servers having same capacity of size, A,B,C,D and E to their relative loads are 90 % , 80 %, 70 %, 60 %, 50 % respectively. To make them as perfect load balancing all the servers into centralized and create new virtual servers F and G to balance the excess traffic. In our example, the load from A and D servers we migrate into F server, B and C servers we migrate into G server and no need to migrate with other server to E. For distribution and new virtual server created and improve the QoS for access from nodes.

Scenario 3: In few cases consolidation workload of 'X' nodes are consumed more energy and 'Y' nodes are unused so no need to allocate the energy to them. To balanced the workload, we make them as centralized and distribute workload according to their priority. For example we consider five identical server having same capacity of size

A, B,C,D,E and E to their relative loads are 70 %, 50 % 30 % from' X ' D and E are empty loads from Y. for each servers. We have distributed the all servers into centralized and migrate the server with appropriate server. In our example, the load from C will migrate to A , B no need to migrate with other server and remaining D and E server we can switch off . To distribute load and switch off the servers which is not in used will improve the QoS and minimized the energy.

Job assignment to server
Initial Assignment:
Stable =0 and gain = 0
While (stable =0)
{
for (s € Si)
{
for (servers considered for resource assignment to rack)
{for (α_{ik} = 1granularity of alpha to 1)
Resource share from Kkt conditions ;
} }
for (s € Si packed based on ranking matrix) {
x= granularity of alpha
y= number of server considered for resources assignment to racks
for(y=1 to Y) {
for(x=1 to X) {
P[x,y] = infinity
for(z=1 to x) {
p[x,y] = max(p(x,y), p(x-y,y-z) partial gain from allocation y th server and α S ik = z)) }
if (client is good performance)
{ calculate the each server usage;
if (client is not good performance)
stable =1
}
else
stable =0;
}

Definition 1: In different users are requested to the server can be defined as matrix. Let M be an l X k dimensional matrix, we defined columns as different resources and rows as VM type for each user needs.

$$M= \begin{bmatrix} m1 \\ m2 \\ m3 \\ .... \\ mn \end{bmatrix} = \begin{pmatrix} \overrightarrow{m11} & \overrightarrow{m12} & \overrightarrow{m13} & \overrightarrow{m1k} \\ \overrightarrow{m21} & \overrightarrow{m22} & \overrightarrow{m23} & \overrightarrow{m2k} \\ ..... & ..... & ..... & ..... \\ \overrightarrow{mn1} & \overrightarrow{mn2} & \overrightarrow{mn3} & \overrightarrow{mnk} \end{pmatrix} \text{------(1)}$$

The aim of the resource allocation problem is distributed resource allocation and mapping from resources to each cloud users to create required virtual machines.

Definition 2: For Virtual server V a possible resource allocation can be describe in allocation matrix $P^{(v)}$:

$$P^{(v)} = \begin{pmatrix} \overrightarrow{p1} \\ \overrightarrow{p2} \\ \overrightarrow{p3} \\ ... \\ \overrightarrow{Pn} \end{pmatrix} = \begin{pmatrix} p11 & p12 & ::: & p1n \\ p21 & p22 & ::::: & p2n \\ p31 & p32 & ::::: & p3n \\ :::::: & :::::: & :::::: & :::::: \\ ps1 & ps2 & ::::: & psn \end{pmatrix} \text{--------(2)}$$

Where p_{ij} define the amount of resource j on virtual server V allocated and i define cloud user.

An allocation decision matrix define P^V all possible resource allocation from physical server.

Definition 3: To increase allocate the resource to 'A ' user without decrease allocation of resource to 'B' another user. Given resource requirement matrix M and sum of the total resource allocation of all physical servers P.

P= $(\sum p_1^{(m)}, (\sum p_2^{(m)}, (\sum p_3^{(m)}$........................ $(\sum p_j^{(m)}, (\sum p_k^{(m)}$. -------------------------- (3)

In multiple resource environment to increase resource allocation to each user and improve resource consumption in each resource in data center. In our proposed approach to improve the resource utilization for virtual servers based on two scenarios.

(1). Max-Min approach: Maximum the minimum energy consumption among the multiple resources of each virtual machine and also physical machine.

(2) Optimization approach: To improve the utilization of physical server to switch off the unused virtual servers.

Definition 4: For improving the bottleneck resource consumption from different types of resource.

Consider vector $V_i^{(n)}$ = ($v_1^{(n)}$, $v_2^{(n)}$, $v_3^{(n)}$, ...$v_k^{(n)}$, $v_{k+1}^{(n)}$...... $v_m^{(n)}$. (4)

It represents the basic resource allocation of physical server with minimum resource utilization 'm' and without creation of virtual machine on physical server.

$$X_{\min}^{(m)} = \min_j \{x_j^{(m)}\} = \min_j \{1- \frac{p_j^{(m)}}{v_j^m} -- \frac{\sum y_{ij}^{(m)}}{v_j} \quad \text{-(5)}$$

DRAOF (Dynamic Resource Allocation Optimization Framework) Dynamic Resource Allocation Optimization Framework is to minimize the energy consumed by using the scheduling proper resource allocation methods and maintain the data centers as required to the workload . In this Framework we defined resource /priority scheduler and calculate the workload, each user send request is received at the data center by the front end server, it calculate the bandwidth of the capacity. The resource scheduler minimum the number of request based on the bandwidth, a) bandwidth request is greater than the data center capacity and no possible to saving energy, b) bandwidth request is greater than the data center capacity, we can find the optimization of number of server and network resource. By using Mixed Integer Liner Programming (MILP), to solve the resource optimization we take two input values.

(a) Throughput demand is calculated by using workload calculator
(b) To calculate oversubscription ratio in data center network.

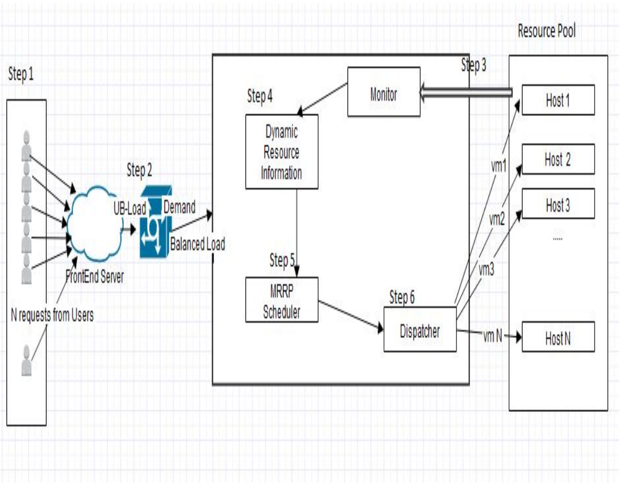


Figure 4 :DRAO Framework

Data center flow model : Let us represented flow model as a directed graph G (N, L) with set of nodes N and set of links (i ,j) □ L, link capacities Mij. Let assume N_{server} and N_{switch} be the subsets N, k € K user demands to send request source servers and it satisfies all user demands. To switch the resource servers to core switches sources to an optimization minimizes the cost of links. The Frond End servers check resource allocation updates by using monitoring the Virtual machines (Server, Network and Storage) dynamically and in few cases the resources sharing as per the user demands. Monitoring helps to

1) Energy Effective Resource Allocation
2) Identification the peak energy efficiency levels
3) Detecting server energy efficiency levels to hosted applications
4) Tracking the failure servers and hanging resource application status.

Data Center Level: Let us consider the data center is connected in form of hierarchical structure which aims that the server capacity and how many virtual machines is using in the physical servers capacity during VM migration. We assume a data center consists of 'N' number of physical machines that are connected by a network, having set of connected links. For each Physical machine consists of two or more virtual machines. To communicate and mapping the Physical machine and Virtual machine we used Virtual Connect Network Placement (VCNP). Let us consider Physical machine(PM) = { $PM_{i\,,}$ i = 1,2,3....n } set of n physical machines in a data center and each physical machine has set of virtual machines VM = { VM_i i= 1,2,3,...n $_{vm}$ } . To allocate each Physical machine we define the set of resource R = {Ri, i =1, 2, 3 ...n $_r$}, such as server capacity, CPU usage, Memory allocate, disk storage capacity. To allocate the total capacity of resource to the Physical machine is denoted by PMtra, the usage of the resource capacity in physical machines is denoted by PMur and also to calculate the available resource to server is denoted by PMar. The total capacity of resource allocation is less than the sum of the usage of the resource allocation and available resource allocation to the servers.

PMtra < PMur + PMar (6) Energy Consumption : Data centers are most expensive to consume huge amount of energy in different location across the world .The energy efficient resource allocation algorithm is how to balance energy consumption and optimize the performance of data center, the performance of the datacenters is depends on the usage of the servers.

$$\text{Energy Efficiency} = \frac{\text{Consumption Energy}}{\text{Total Energy}}$$

In order to calculate for a particular system, the total energy consumption is sum of its individual components, energy consumed by the users and energy consumed in switch (network) and last energy consumed in data center (LCDC).

$E_{Total} = E_{user} + E_{switch} + E_{LCDC}$

Energy consumption of nodes in datacenter consists of consumption of CPU's in servers, disk storage and network devices. To compare to another system resource CPU's servers consumes huge amount of energy we can define as

$E(u) = (E_{max} - E_{idle}) * C/100 + E_{idle}$

E_{max} is the maximum energy consumed and server is maximum utilized , E_{idle} is the minimum energy consumed and server is not running or unutilized , C is defined the energy consumed by the server in the CPU utilization

Algorithm: Modified Round Robin + Priority Scheduling (MRRP)

Input: Initialized processing units and power using DVFS = {dv1, dv2, dv3, dv4... dvn} and number of jobs J_i = (i=1, 2, 3...n)

Output: To assigned the VM machines with minimum energy capacity of data centers

Step 1: Initialized the list of Servers with same capacity of size and memory.

Step 2: Share the Virtual machine to specify physical machines with in interval time

Step 3: Sort the Physical machine and Virtual machines in decreasing order of their energy consumption PM_i ={i = 1,2,3,.....m } and VM_j = {j = 1,2,3,4.....n }

Step 4: Assigned the VM as per the user demands to his Physical machines, and number of tasks T_k = {k =1, 2, 3 ...p}

Step 5: for each J_i do

Step 6: if number of priority machines in m in any rack >= number of p in J_i then

Step 7: Give the priority the user demands PM and VM

Step 8: for each Power_ level do

Step 9: assigned the each VM using Round Robin allocation

Step 10: Continue Step 7

Step 11: if VM is assigned as per the user demands

Step 12: calculate the number of VM and used and remaining unused VM is in inactive mode

Step 13: break
Step 14: end if
Step 15: if VM allocation size is greater than the allocate size in servers
Step 16: Create Virtual machines as per user demands in particular interval of time (temporally)
Step 17: else VM placement size is less than the allocate size in server
Step 18: Switch off the unused servers
Step 19: end if
Step 20: End

Online Virtual Machine Placement: In this section, we work on the online Virtual Machine Placement; we get the information by predicting the VMs set to priori based on the past information and allocate the each VM's resource demand and total number of VM requests. Suppose a in server having M physical machines (PM's) having the capacity of 'K ', and it is denoted by C_k. In this we have 'N' VMs waiting for the placement based on the user demand 'i' VM is R_i. Scenario 1: Let us consider 3 Physical machines and each PM having 3 VM capacity but allocate only two VM's and remaining 1 VM is unused for each physical machines, so we used another physical machine to move the unused Virtual machines into PM4 and also dynamically inactive the unused capacity of each physical machine.

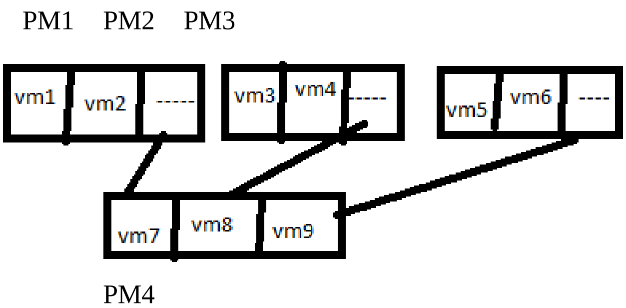


Figure 5 : VM Placement

Scenario 2: Let us consider 3 physical machines and each PM having the capacity of 3 VM, but it resource allocate only 2 VM; s and remaining 1 VM space is unused for each physical machine . We can move the nearest physical machine which is resource is available from PM2 to PM1 and also PM3 to PM2. So that we can switch off the physical machines and allocate the resource effectively.

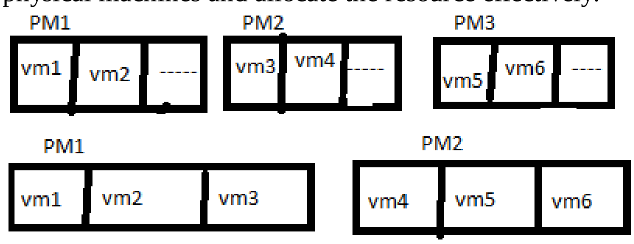


Figure 6 : VM Placement 2

Scenario 3: Let us consider, Location A, 4 Physical machines having 3 VM's capacity and but location A less capacity of VM's compare to user demands. So we allocate resource which is nearest location B those are not in active mode. As per the user demands to active the VM's in location B. Example, in location A , user demands more than 4 VM's , sorting the activating the VMs in PM1,PM2,PM3 in location B.

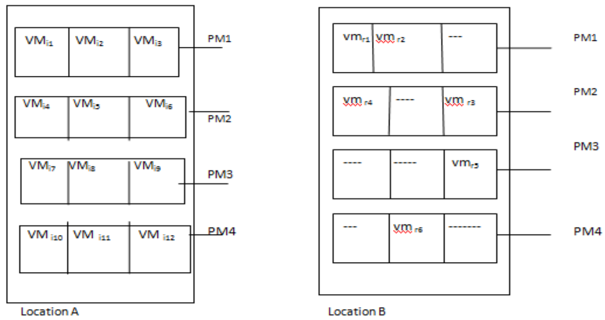


Figure 7 : VM Placement 3

Algorithm: VM Placement Algorithm.
Input: Initialized list of Physical machines and assigning the VM's for each server
Output: Allocate list of VM's as per the User demands
Step 1: Initialized the list of PM's with same capacity of size and memory and assign the VM's.
Step 2: Sort the all PMs for each servers.
Step 3: for i =1 to N does
Step4: sort PM's increasingly by t_{ij}
Step 5: for j=1 to M do
Step 6: if C_k is sufficient for VM_j then
Step 7: replace or move VM_j in PM_i
Step 8: if VM_i capacity > Placement Capacity,
Step 9: Do Replace the VM_i
Step 10: else reject the VM_i
Step 11: end loop
Step 12: End

Dynamic Virtual Machine Migration (DVMM) : In a single physical machine we can create multiple virtual machines, so it is increase resource utilization. To reduce the energy consumption we used the switching idle nodes to lower power modes (sleep and hibernation modes). By using Dynamic migration the VMs can be dynamically integrated to the minimal number of physical nodes with respected resource needs. In this scenario, Virtual machine is moving from the existing placement to other new placement for live migration for different values are placed on different hosts. The main aim of the algorithm is to increase the VM write functionalities of pre- copy in memory and it automat ally schedules the cpu time for assigned virtual machines, so that dirty rate should be decrease. We consider a VM running program have x % of memory occupied and the dirty rate, execution speed based on the priority levels. So VM 's would prefer to schedule the processes with higher priority and remaining other process with normal priority will be schedule and executed. With high priority of dirty rate and `memory size of VM's will decrease than the low priority of VM's. We modified the live migration of VM, migrated each VM's optimized and utilized the maximum number of memory and separate the dirtied memory in VM's and continues until reaches the threshold point . After reached the threshold point we schedule as low priority and make as null.

Algorithm : Optimized Live migration VM.

Step 1 : Initialized VM's , Memory Size, Dirty memory VM's.
Step 2: Memory Size ← VM's memory
Step 3 : Dirty memory ← 1
Step 4 : While Size of Memory size > Threshold point and Dirty memory < Temp
Step 5 : Continue to calculate VM's dirtied memory
Step 6 : Memory size ← Temp
Step 7 : Read current CPU allocated for M to E
Step 8 : Each VM , Schedule E % of the CPU
Step 9 : Execute Each VM based on Priority schedule.
Step 10.: End While
Step 11 : End

For a given task on selected resource, Maximum utilization of resource utilization is designed based on the prioritize resource sharing for a given resources and remaining tasks running in parallel is superior than the energy consumption of a task

Simulation Results

In our algorithm, we used the GreenCloud Simulator[17] to calculate energy of data center, we used two resource optimization approaches and implementation of DRAOF. In this simulator supports different tired data center architectures and also incorporate detailed energy models of data center resources. MRRP(Round Robin + Priority Scheduling) is used to solve MCMCF optimization in data center and adopting novel approaches at 100 % CPU utilization and consume maximum energy.

Energy consumption parameters :

The parameters for energy consumption calculation from the different levels of servers and switch energy and we simulated on Unix operation system of 2GB RAM . In simulation we make core as increases 5 each level of servers and users as constant, TE (Total energy), S.W.E (Switched energy Core, Aggregations and Access) Energy Consumption in Data Centers From above table we have compare the different types of existing algorithms, and proposed algorithm RRPS. Compare between the Existing and Proposed Algorithm

Table 1: Power Consumption

Parameters	Power Consumption			
	BFA	RRB	PS	**MRRP-Proposed**
DataCenter	512.4 K	417.5 K	380.9 K	209.5 K
Server	415.6 K	353.7 K	279.2 K	172.1 K
S.W.E Core	58.2 K	48.7 K	69.9 K	26.6 K
S.W.E Access	38.6 K	15.1 K	31.8 K	10.8 K

Table 5 : Simulation results using Greencloud Simulation

Server	No.of Tasks	T.E	S.W.E Core	S.W.E Access	S.E
20	344	932	274.4	26.6	82.1
40	688	2010.2	597.6	53.2	164.1
60	1038	3234.9	969.6	79.8	246.3
80	1365	4605.2	1390.3	106.4	327.9
100	1709	6121.9	1859.7	132.9	409.9
120	2069	7785.6	2377.9	159.5	492.4
140	2380	9594.3	2944.9	186.1	573.6
160	2651	11548	3560.9	212.7	653.6
180	3052	13651.5	4225	239.3	737.2
200	3460	15901.6	4938.2	256.9	821.1

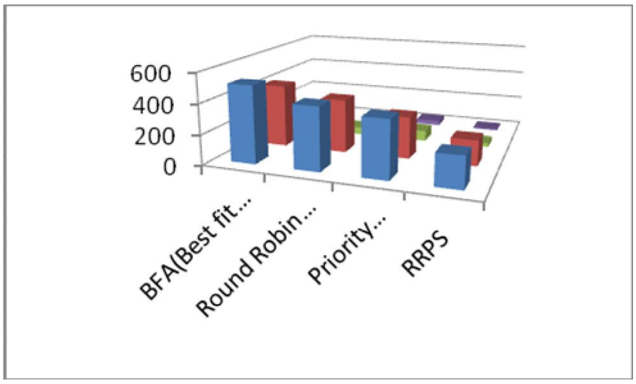


Figure 8: Energy Consumption in Data Centers

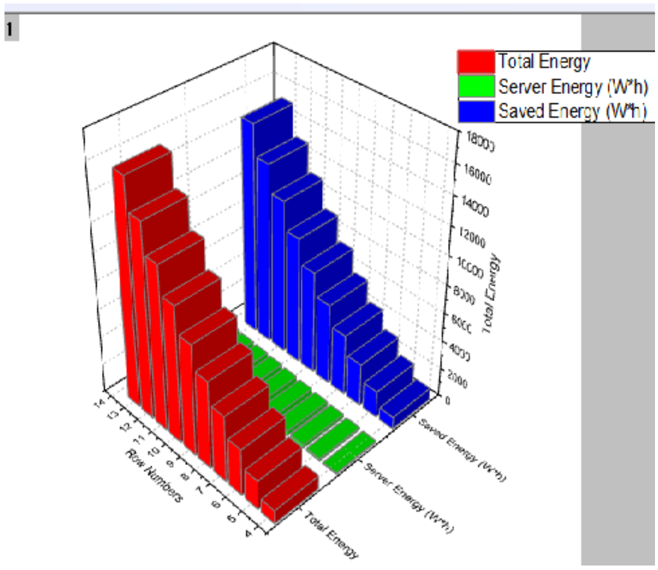


Figure 9 : Utilization of Energy

From above graph we can observed, minimum energy is consumed for servers and it distributes loads equally among servers and switches, so that we can network traffic is balanced and no servers is overload. Energy Consumption in Data Centers

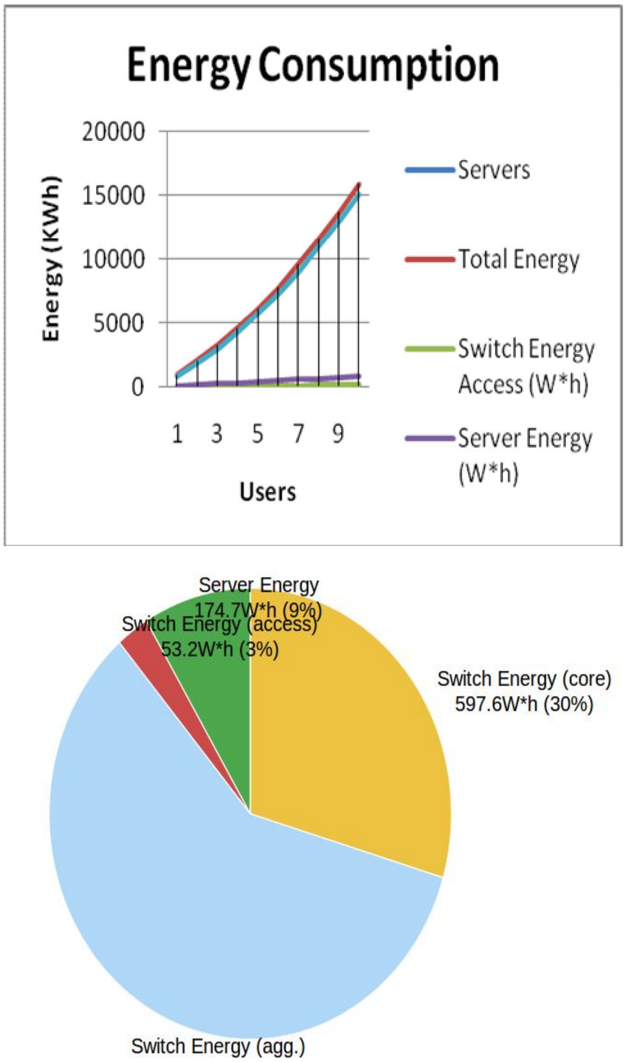
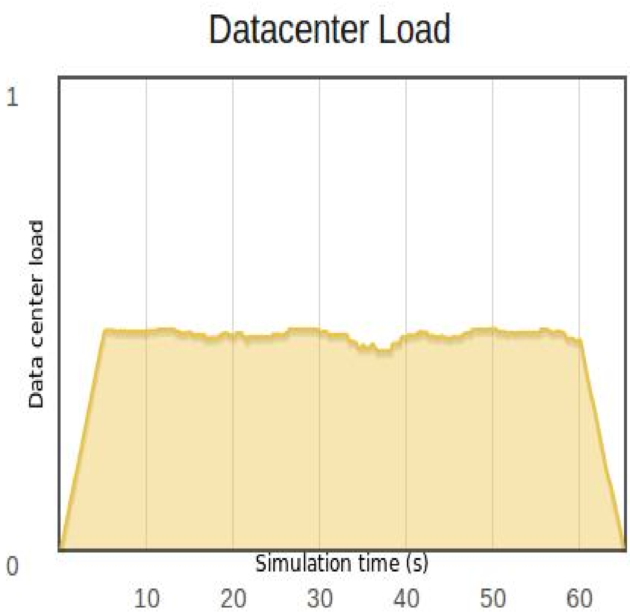
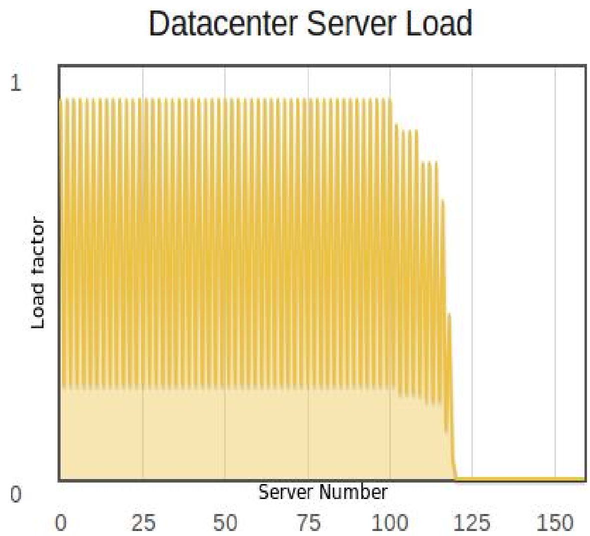
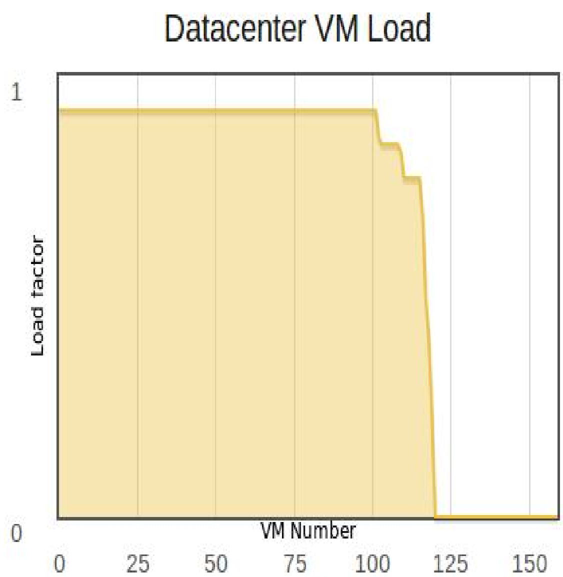
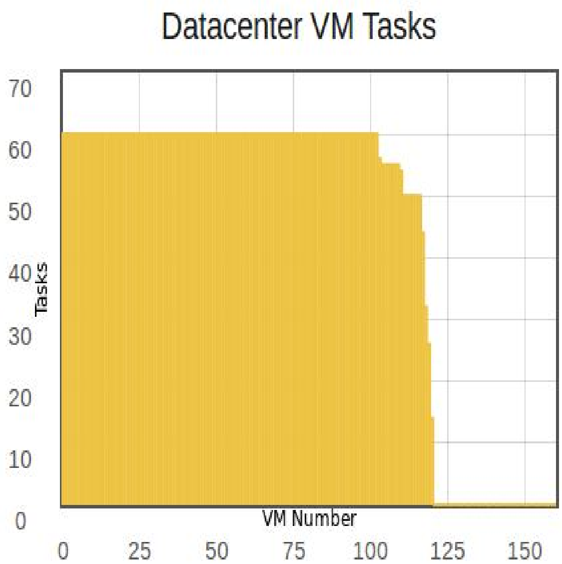


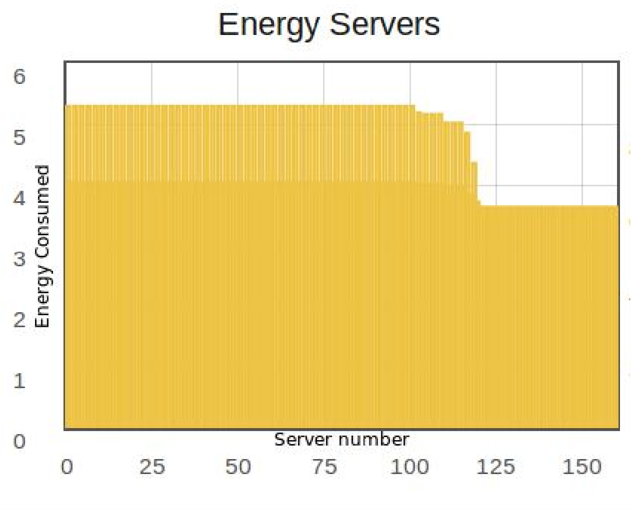
Figure 7 : Different areas of energy consumption in data center











5.CONCLUSION

In recent years power usage is increased in data centers In this DRAOF (Dynamic resource allocation optimization framework) minimize energy consumption with maximum utilization of resource allocation .However we have consider energy efficiency and together QoS and load balance , based on their performance we incorporate energy efficiency in resource allocation scheduling. DRAOF works saves more energy as compared to other frameworks and it reduced average end-to- end delay by using RRPS, it is depends on the number of resources is usage. We also planning in future work to development of the power aware scheduling algorithm using genetic methods and also to work in real time applications for example Sparse Cholesky decomposition.

6.ACKNOWLEDGMENT

This research was supported by the Computer Science and Technology in Sri Krishnadevaraya University, Anantapuramu, Andhra Pradesh.

REFERENCES

[1] M. Creeger, "Cloud Computing: An Overview," Queue, vol. 7, no.5, pp. 2:3–2:4.

[2] M.Aleksy, A.Korthaus, and M. Schader, " Design and implementation of flexible load balancing service for CORBRA- based applications,". in PDPTA;01: USA: IEEE Computer Society, June 2001.

[3] Beloglazov, A., Buyya, R.: Energy efficient resource management in virtualized cloud data centers. In: Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on, pp. 826–831 (2010).

[4].Heller, B., Seetharaman, S., Mahadevan, P.,Yiakoumis,Y., Sharma, P., Banerjee, S., Mckeown, N.: Elastictree: Saving energy in data center networks. In: IN NSDI (2010).

[5] Y. Wang, R.J.T. Morris, Load sharing in distributed systems, IEEE Trans. Comput.34 (3) (1985) 204–217.

[6] Dewitt, D. and Stonebraker, M. MapReduce II blogpost; http://databasecolumn.vertica.com/ database-innovation/mapreduce-ii/

[7]. Katoh, N., Ibaraki, T.: Resource allocation problems. In:Du, D.-Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization, vol. 2, pp. 159–260. Springer (1998).

[8]. J. von Neumann and O. Morgenstern, Theory of Games and Economic Behavior, Princeton University Press, 1953.

[9]. S.U. Khan and I. Ahmad, "'Non-cooperative, Semi-cooperative, and Cooperative Games based Grid Resource Allocation," Int'l Parallel and Distributed Processing Symposium, 2006.

[10]Khan, S.U.: A game theoretical energy efficient resource allocation technique for large distributed computing systems. In: International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA), pp. 48-54. Las Vegas (2009).

[11].Khan, S.U., Min-Allah, N.: A goal programming based energy efficient resource allocation in data centers. J. Supercomput (2011). doi:10.1007/s11227-011-0611-7.

[12]. Liu L, Wang H, Liu X, Jin X, He WB, Wang QB, Chen Y. GreenCloud: a new architecture for green data center. In: 6th International Conference on Autonomic Computing and Communications, Industry Session; 2009 June 15–19. Barcelona, Spain: ACM; 2009. pp. 29–38.

[13]. Pinheiro E, Bianchini R, Carrera EV, Heath T. Load balancing and unbalancing for power and performance in cluster-based systems. In: Workshop on Compilers and Operating Systems for Low Power, Volume 180. Citeseer; 2001. pp. 182–195.

[14].Google Data Center Efficiency (online). http://www.google.co.in/about/datacenters/efficiency/internal/

[15]. Kwok, Y.K., Hwang, K., Song, S.: Selfish grids: Game-theoretic modeling and nas/psa benchmark evaluation. IEEE Tran. on Parallel and Distributing Systems 18(5), 1–16 (2007).

[16]. Available at http://www.green500.org/.

[17]. Available at http://greencloud.gforge.uni.lu/

[18] Rajkumar Buyya, James Broberg, Andrzej Goscinski (2011), "CLOUD COMPUTING:Principles and Paradigms " , A Jhon Wiley & Sons, Inc. Publication

AUTHOR

C. Vijaya Kumar is currently a Ph.D, Student in the Department of Computer Science and Technology from Sri Krishnadevaraya University. He received the B.Sc degree in Computer science from Sri Krishnadevaraya University and M.C.A from the Sri Venkateswara University Tirupathi, in 2010, research interests include Cloud Computing, Hadoop and Data mining.

Dr.G.A Ramachandra is an Associate professor at Sri Krishnadevaraya University. His research interests are Data Mining, Computer Networks and Cloud computing.