Optimal Energy Efficient Scheduling based on VM Energy Consumption in Cloud

M Dhanalakshmi
Dept of CSE
East Point College of Engineering & Technology
Bangalore, India

Anirban Basu
Dept. of CSE-R&D Centre
East Point College of Engineering & Technology
Bangalore, India

ABSTRACT

Cloud computing has emerged as a flexible and powerful computing platform which offers different types of services to users. However, the energy consumption and carbon emission in cloud data centre have massive impact on global environment triggering intense research in this area. Growth in demand for cloud based services is resulting in increasing of utilization of cloud resources and consequent increase in the energy consumption. While there is a need to maximize utilization of the resources, the energy consumption needs to be reduced. Therefore we need to determine the operational optimal point which reduces the energy consumption for a desired value of resource utilization.

In this paper, we propose a technique to find such an optimal operational point in Cloud computing environment. The technique is based on measuring the energy consumption of VMs created for a task, classifying the VMs based on their energy consumption and resource utilization, and then performing scheduling of VMs. The time efficiency of this algorithm is$f(n) \in \theta(mn)$.

The effectiveness of the proposed technique has been verified by simulating on CloudSim. Experimental results confirm that the technique proposed here can significantly reduce energy consumption for the desired value of resource utilization.

Keywords

VMs (Virtual Machines), EDU (Energy Distribution Unit), PDU (Power Distribution Unit).

1. INTRODUCTION

High energy consumed by computing and network systems, in large data centres with high capacity servers and storage is causing serious concern to the environmentalists [1], [2]. It was found that the energy consumed by Google was 2,675,898 Mega Watt Hours [3]. Energy consumption in any resource can always be reduced by decreasing the utilization i.e., by switching off or putting in sleep mode when not in use. But when the servers are being used we can minimize energy consumption by using energy efficient techniques as discussed in [4]. However most of these techniques have not considered increasing the resource utilization to desired value while minimizing the energy consumption when providing the services on the cloud. In this paper, we propose a technique to find such an optimal operational point in Cloud computing environment.

Cloud Computing platforms [5] generally use Virtualization. Virtualization of computer system resources includes processor, memory and I/O devices. Virtualization techniques enable multiple tasks to be consolidated [6] on few servers and results in reduced idle power cost thereby minimizing energy consumption.

Earlier in the data centre the energy measurement of physical server was done by built in energy meters. Several methods were used in servers to measure energy consumption at the outlet using Energy Distribution Units (EDUs). The disadvantage of such measurement is that it increases the energy consumption by the equipment attached and this leads to heat dissipation. The hardware based solution providing energy usage statistics are therefore unsuitable in data centre where low energy consumption is desired.

However power consumption in the VMs cannot be measured purely by hardware equipment. A number of alternative methods of energy measurement for Virtual Machines have been discussed [7]. However, this would require vendor support and is not applicable to systems where their hardware does not support this kind of fine-grain monitoring of individual components.

In this paper, we consider a virtualized data centre and propose an optimal method of scheduling which takes into account the resource utilization and energy consumption. An operational optimal point is determined where the energy consumption is minimized and resources utilization is maximized.

In the proposed technique tasks arrive at the data centre and VM is created for each task. The VMs are assigned to the physical servers arbitrarily for the purpose of determining the energy consumption of the VMs [8], [9]. The energy consumption of each VM is measured from its resource utilization by using power equations similar to that proposed in [10], [11]. The resource usage is converted to energy usage based on energy equation of the resources. Next, the VMs are classified into three types based on the resource usage as CPU intensive, I/O intensive or Network intensive [12]. Finally the VMs are scheduled onto the physical servers [13] taking into account the resource utilization and energy consumption of the server such that server utilization is increased but energy consumption is minimal [14], [15]. The technique was simulated on CloudSim [16] and experimental results show the algorithm effectiveness of the proposed technique.

The paper is organized as follows. Section II we discuss Related Work. Section III describes System Scheduling Architecture Model. Section IV is the Proposed Techniques. Section V is Experimental Results. Section VI Concludes the paper.

2. RELATED TECHNIQUES

Monitoring and minimizing energy consumption of the Cloud System is difficult using hardware Power Distribution Unit (PDU) as they provide less accurate values on power usage. An alternative solution is required, as noted by Stoess et al. [7] who discussed the possibility of energy measurement for Virtual Machines, which cannot be done by connecting hardware measurement devices. Stroess et al. proposed that power usage of the hardware device is reported to device driver, which in turn send the power usage information to operating system. However this would require vendor support.

Aman Kansal et al. [8] in their paper author present a solution for VM power metering; Virtualization is often used in cloud computing platforms for its several advantages in efficiently managing resources. However, virtualization raises certain additional challenges, and one of them is lack of power metering for virtual machines (VMs). VM power cannot be measured purely in hardware. Authors in their paper present a solution for VM power metering, named Joulemeter. They build power models to infer power consumption from resource usage at runtime and identify the challenges that arise when applying such models for VM power metering. Their approach is designed to operate with extremely low runtime overhead while providing practically useful accuracy.

S. Rivoire et al. [9] have considered the common infrastructure to fit a family of high-level full-system power models and compare these models over a wide variation of workloads and machines, from a laptop to a server. Their analysis shows that a model based on OS utilization metrics and CPU performance counters is generally most accurate across the machines and workloads are tested. It is particularly useful for machines whose dynamic power consumption is not dominated by the CPU.

Husain Bohra et al. [10] in their paper describe comprehensive online monitoring of resource utilization along with the implementation of power-aware policies to reduce the total energy consumption. Many of the existing techniques provide online power monitoring based on the power consumption of a physical node running one or more Virtual Machines (VM). They lack a fine-grained mechanism to profile the power of an individual hosted VM. In their work authors present a novel power modelling technique, VMeter, based on online monitoring of system-resources having high correlation with the total power consumption. The monitored system sub-components include: CPU, cache, disk, and DRAM. This model predicts instantaneous power consumption of an individual VM hosted on a physical node besides the full system power consumption.

James W. Smith et al. [11] they developed Cloud Monitor, an open-source, automated, scalable, resource and power-usage reporting tool can provide fine grain utilization information. The tool includes a power model generation facility to provide software-based energy measurements based on resource usage. This facility provides an accurate estimation of power consumption that can be used to precisely calculate the financial cost of electricity to power a deployment, and opens the possibility of accurate forecasting for cloud providers that may in the future apply a separate tariff for energy consumption.

James W. Smith et al.[12] in their paper the effect of different workloads on server power consumption in a Private Cloud platform is discussed and display a noticeable difference in energy consumption when servers are given tasks that dominate various resources (CPU, Memory, Hard Disk and Network). .

Sobir Bazarbayev et al. [13] in their work, authors discussed the Infrastructure as a Service (IaaS) clouds, provides on demand nature of services. In the large cloud data centre hundreds of thousands of virtual machines can be deployed and terminated. The content based scheduling algorithm schedules the VMs with high content similarity on the same hosts. When deploying a VM on a destination host reduces the amount of data transferred. The content based scheduling algorithm reduces the network traffic.

However none of the methods are not accurate as they do not consider all hardware components in calculating energy consumption. The proposed techniques are more accurate, because the measurement of VMs is based on OS utilization metrics and CPU performance counters and the measurement of energy consumption includes all the sub components of Networks and switches etc. Further by classification and scheduling technique we find the operational optimal point at which energy consumption is minimized and resources utilization is maximized.

3. SYSTEM SCHEDULING ARCHITECTURE MODEL

The optimal VM scheduling is based on measurement of VM energy consumption and resource utilization. The system scheduling architecture model is as shown in Figure 1 is used to perform optimal scheduling.

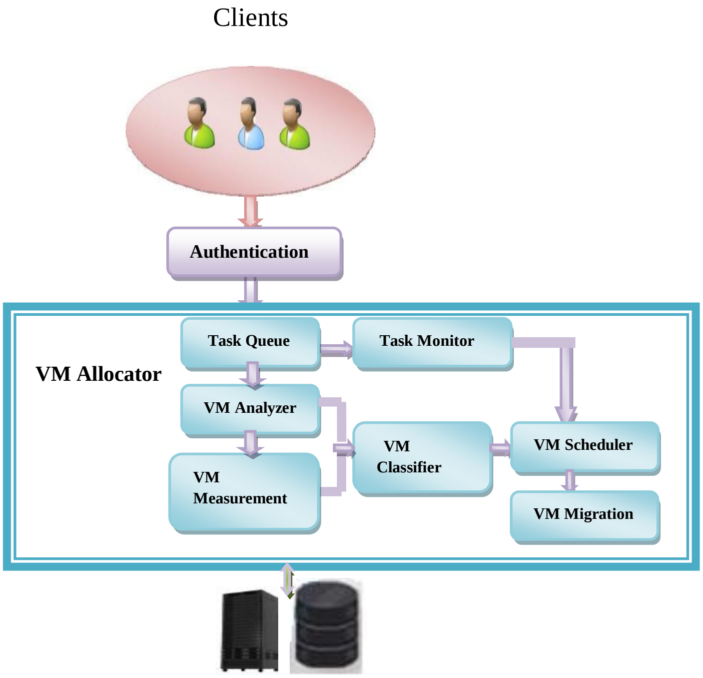

Figure 1. System Scheduling Architecture Model

In the cloud environment, the clients send the request to execute their tasks on cloud data centre. After authentication the client gathers the information about the resources and existing cloud VM status. The VM allocator is an interface between the cloud system resources and clients. It consists of components to design and support for system scheduling architecture model.

The Task Queue is sub component of the architecture receives the tasks send by the client; this Task Queue preserves the priorities of the incoming tasks. The Task Monitor monitors which server machines in the data centre is power on/off and determines the existing running VMs energy consumption information on each server.

VM Analyzer analyzes the task submitted by the clients based on that VMs are created arbitrarily. VM Measurement measures the VM energy consumption and gives this information to VM Classifier.

The VM Classifier receives the VMs information from the VM analyzer. It classifies the VMs into three types CPU intensive, I/O intensive or Network intensive. VM Scheduler receives the classified VMs from the VM Classifier and receives the existing VMs energy consumption information of the server from the Task Monitor. It schedules the VMs and performs optimal scheduling. It also decides when VMs are to be added or removed to meet the demand. The VM Migration performs migration across physical machines to minimize energy consumption.

4. PROPOSED SCHEDULING TECHNIQUE

In this paper we develop a scheduling technique in order to find the optimal operational point at which maximizes the resource utilization while minimize the energy consumption. The proposed optimized energy efficient algorithm is an NP hard problem. In the paper a different approach is taken and the greedy algorithm is used which yields an optimal solution. The time efficiency of this algorithm is $f(n) \in \theta(mn)$. the proposed technique is implemented in three stages as discussed below.

4.1 Measurement of VM Energy Consumption using Energy Equations

The tasks arrives the data centre, for each task the VM is created and arbitrarily allocated. The energy usage of individual VM is measured by monitoring the energy consumption of VM resources on server namely CPU, cache, DRAM, Hard disk, Network components and switch components. The energy equation is based on OS utilization and performance counters [9]. Because of the lack of measurement of any direct hardware performance counter. The hardware architecture generates the resource utilization per VM about CPU, Cache, DRAM and Disk through the system events such as: CPU-CLOCK-NOT-HALTED, RAM-ACCESSES, INSTRUCTION-CACHE-READ, and DATA-CACHE-READ via performance counters register [10], [11]. These events are used to monitor the power consumption of the VM.

The power equation based on the hardware resources is given in equation 1 gives better accuracy as it includes power consumption of the networks sub components (network interface cards etc.) and switch components(the chassis, line cards, ports).

$$P(total)=k_0+(k_1*P_{cpu})+(k_2*P_{cache})+(k_3*P_{dram})+(k_4*P_{disk})+(k_5*P_{netcomps})+(k_6*P_{switchcomps}) \quad (1)$$

The values for k_0 (constant power during calibration) and k_1 (for CPU), k_2 (for Cache), k_3 (for DRAM), k_4 (for Hard Disk), k_5 (for Network), k_6 (for Switch Components) are obtained by measurement.

Where the P_{cpu} = CPU utilization $_,$ P_{cache} = Cache memory access count$_,$ P_{dram} = DRAM memory access count $_,P_{disk}$= Hard disk I/O rate$_,$ P_{net} = Network I/O rate$_,$ $P_{switchcomps\ =}$ Switch components rate.

We define the total system power as a function of the utilization of CPU, Cache, DRAM, Hard disk, Network components and Switch components. If Psysmax is the maximum power consumed when the server is fully utilized; k is the fraction of power consumed by the idle server (e.g. 70%); u is the CPU utilized, r is DRAM access count, d is Hard disk I/O rate, c is the Cache memory access count, n is the Network I/O rate and s is switch components rate then the total system power is as given below:

$$Psys(u+r+d+c+n+s)=k.Psysmax+(1-k).Psysmax.u.r.d.c.n.s \quad (2)$$

Psysmax is normally 250 W, which is the normal value for modern servers. The utilization of the CPU, DRAM and Hard disk, Cache, Network components, Switch components may change over time due to workload variability. The CPU, DRAM, Hard Disk, Cache, Network components, Switch Components utilization is a function of time and is represented as u(t), r(t), d(t), c(t), n(t), s(t). Therefore, the total energy consumption by a physical node (E) can be defined as integral of the power consumption function over a period of time from t_0 to t_1 as shown in (3).

$$E = \int_{t0}^{t1} P_{sys}\big(u(t)+r(t)+d(t)\big)+c(t)+n(t)+s(t))dt \quad (3)$$

4.2 VM Energy Consumption Analysis and Classification

The proposed method is based on classification of VMs [12] for the purpose of determining the operational optimal point which reduces the energy consumption for a desired value of resource utilization.

The VMs are classified into three types based on the resource usage: CPU intensive (where CPU and cache usage is dominant. Hard disk and DRAM consume negligible energy), I/O intensive (where Disk usage is dominant) and Network intensive (where Network components usage is dominant). The power consumption in the three types of VMs based on usage of hardware is referred to as Pcpuintensive, Pi/ointensive and $P_{Networkintensive}$ and given by the following equations in (4), (5) and (6)

$$Pcpuintensive = k_0+ k_1*Pcpu+ k_2*Pcache \quad (4)$$

$$Pi/ointensive = k_3+ k_{4*}P_{DRAM}+ k_{5*}P_{disk} \quad (5)$$

$$P_{Networkintensive} = k_6+ k_{1*}P_{cpu}+ k_{2*}P_{cache}+ k_{7*}P_{network}+ k_{8*}P_{switch} \quad (6)$$

The values of $k_{0,}$ $k_{3,}$ k6 (constant power during calibration) and k_1 (for CPU), k_2 (for Cache), k_4 (for DRAM), k_5 (for Hard

Disk), k_7 (for Network), k_8 (for Switch Components) are obtained by measurement.

Where P_{cpu} = CPU utilization, P_{cache} = Cache memory count, P_{dram} = DRAM memory count, P_{disk}= Hard disk I/O rate, P_{net} = Network I/O rate, $P_{switch =}$ Switch Components rate.

The corresponding VMs energy equation is given by equation (7), (8) and (9).

$$E_{cpuintensiv} = \int_{t0}^{t1} P_{cpuintensive} dt \qquad (7)$$

$$E_{i/ointensiv} = \int_{t0}^{t1} P_{i/ointensive} dt \qquad (8)$$

$$E_{Netintensive} = \int_{t0}^{t1} P_{Networkintensive} dt \qquad (9)$$

The classification helps to achieve maximum utilization of resources while minimizing the energy consumption. For example, to schedule the VM that is CPU intensive, we search for the server whose energy consumption is less than the allowable prescribed maximum energy consumption. Again VM scheduling is based on determining the server in which the existing running VMs must be I/O intensive or VMs can be least Network Utilization. To Schedule Network intensive VM is based on determining the server in which the existing running VMs must be I/O intensive or VMs can be least CPU utilization.

4.3 Proposed Optimized Energy Efficient Scheduling

In the proposed optimized energy efficient scheduling algorithm first we measure the energy consumption of the servers in the data centre as well as the resource utilization of existing VMs on the servers in terms of (CPU, memory, network and Hard disk). These values are stored into a table in ascending order

Secondly the m tasks arrives the cloud data centre. We analyse the task based on that VM is created for each task arbitrarily. The energy consumption of the each VM and their resource utilization in terms of (CPU, memory, Hard disk, DRAM) is measured and these values are stored into a table in descending order.

The VMs are classified into three types based on the resource usage as CPU intensive, Network intensive, and I/O intensive.

For allocating the VMs on the server in this proposed method first the Remaining energy consumption (REC) is initialized to Maximum Energy Consumption (MEC) in each server. Then check each VM energy consumption (VEC) is greater than REC then exit the algorithm if exceeds the prescribed limit. Otherwise check if the VM Type is CPU intensive then check for all existing VMs in the server is not CPU intensive. The existing running VMs can be least Network Utilization. or VMs can be I/O intensive. Then select the VM and decrement the REC by the selected VM energy consumption.

Otherwise check if the VM type is not CPU intensive then check it is Network intensive and check for all existing VMs in the server is not Network intensive. The existing running VMs can be least CPU Utilization. or VMs can be I/O intensive. Then select the VM and decrement the REC by the selected VM energy consumption.

Otherwise check if the VM type is not Network intensive then it is I/O intensive and check for all existing VMs in the server is not I/O intensive. The existing running VMs can be least CPU utilization or VMs can be least Network utilization. Then select the VM and decrement the REC by the selected VM energy consumption.

Finally execute the selected VM on the server which leads to maximize the utilization of the resources while reducing the energy consumption.

The total energy consumption on each server is updated. Remove the executed VMs from the VMList. Schedule the remaining VMs in the VMList to the next server in the data centre if the energy consumption on the server exceeds the prescribed limit. If the VMs in the VMList is not able to schedule on the server in the data centre migrated the VMs to the servers in another data centre. The algorithm is illustrated in Figure. 2 and the following terms are used.

MEC(Maximum Energy Consumption in each server)
REC (Remaining Energy Consumption in each server)
SEC (Server Energy Consumption)
VEC (VM Energy Consumption)
SVM (selected VMs) using optimized energy efficient scheduling.
C (CPU Intensive as specified in Section IV B)
N (Network Intensive as specified in Section IV B)
I/O (I/O Intensive as specified in Section IV B)
VT (VM Type can be ‘C’ or ‘N’ or ‘I’)
CUR (CPU Utilized Rate)
CUU (CPU Under Utilized)
NUR (Network Utilized Rate)
NUU (Network Under Utilized)
IOUR (I/O Utilization Rate)
IOUU (I/O Under Utilized).

The main objective of this algorithm is to determine the operational optimal point at which maximize the utilization of the resources while reducing the energy consumption. The optimal function and its constraint are given in equation (10) and (11).

The optimal function

$$\sum_{i=1}^{n} SEC_i * SVM_i \qquad (10)$$

Subject to the constraint

$$\sum_{i=1}^{n} VEC_i * SVM_i \quad \leq \quad MEC \qquad (11)$$

To obtain optimal solution the tasks are arranged in decreasing order of $\frac{SEC_i}{VEC_i}$

```
1. Input: CDataCentre, CVMList,
        CServerUtilizationList,CVMUtilizationList
2. Output: The Selected VMs for each server to minimize
          energy consumption

3. Foreach CServer in CDataCentre Do
4. CServerUltilizationList.SortAscendUtilization()
```

```
SVM ← 0;
SEC.MeasureSEC ()
Foreach CVM in CVMList Do
CVMUltilizationList.SortDescendVMtype ()
REC ← MEC
While (REC) Do
if (CVM.VEC > REC) break; // exit the algorithm
if (CVM.VT == 'C')
  if (CServer.VT != 'C')
   if (((CServer.VT=='N')&&(CServer.CUR <= 'NUU'
     ))||(CServer.VT == 'I/O'))
     SVM ← 1;
     REC = REC-CVM.VEC
   endif
 endif
else
 if (CVM.VT == 'N')
  if (CServer.VT != 'N')
   if (((CServer.VT=='C')&&(CServer.CUR <= 'CUU'
     ))||(CServer.VT == 'I/O'))
     SVM ← 1;
     REC = REC-CVM.VEC
   endif
 endif
else
 if (CVM.VT == 'I/O')
  if(CServer.VT != 'I/O')
    if((CServer.VT == 'C')||(CServer.VT == 'N'))
   SVM ← 1;
   REC = REC-CVM.VEC
   endif
 endif
endif
 CServer.ExecutetheSVMs();
 CServer.UpdateSEC();
 CVMList.RemoveExecutedSVM();
 RemainingVMs.Allocate(CVMList);
endwhile
endfor
endfor
Migrationlist.Add(CVMList);
```

Figure 2. Algorithm for optimized energy efficient scheduling

This technique determines the operational optimal point which reduces the energy consumption for a desired value of resource utilization as illustrated in Figure 3 and 4

5. EXPERIMENTAL RESULTS

We propose a technique to find an operational optimal point in Cloud computing environment at which maximizes the resource utilization while minimizing the energy consumption. This technique is based on measuring the energy consumption of VMs. Energy consumed by different types of VMs was compared and the result of comparison is as shown in Figure 3. The analysis shows that CPU intensive VMs consume maximum energy consumption compared to Network and I/O intensive VMs. The Network intensive VMs energy consumption is less compared to CPU intensive where as it is more than I/O intensive VMs. The I/O intensive VMs energy consumption is less than Network and CPU intensive VMs.

The experimental result of the optimal energy efficient scheduling is based on resource utilization and energy consumption of VMs. This technique determines the operational optimal point which reduces the energy consumption for a desired value of resource utilization**.** We use the proposed algorithm and schedule the VMs on different percentage of overall resources available and find the suitable operational optimal point at which % of resource utilization is optimal and the energy consumption is minimized. For example we consider 30% of the overall resources and the remaining 70% of the resources in sleep mode next consider 60% of the overall resources and the remaining 40% of resources in sleep mode and then considering 100% of overall resources. Measuring the energy consumption in terms of MWH (Mega Watt Hour) one Mega Watt Hour is one Million Watt Hour.

First we consider 30% of overall resources and 3000 VMs. The algorithm must schedule the VMs on 30% of overall resources. The energy consumption by the VMs is reduced. The total energy consumed is the sum of energy consumed by the VMs scheduled by the algorithm and the power saved by 70% of the resources in the sleep mode.

Second we consider 60% of the overall resources and 3000 VMs. The energy consumption by the VMs is reduced and it is lesser than 30 % of overall resource usage. Because as the % of resource increased from 30 to 60, the algorithm can schedule the VM on the 60% of overall resource by determine more appropriate server that further reduces the energy consumption. The total energy consumed is the sum of energy consumed by the VMs scheduled by the algorithm and the power saved by 40% of the resources in the sleep mode.

Finally we consider 100 % of the overall resource usage the energy consumption is less than the 30% of resource usage but more than 60 % resource usage as illustrated in the Figure 4. In this case 100% of resource is utilized no power is saved by the resources in the sleep mode. The proposed optimal scheduling gives operational optimal point at 60% of resource usage. The implementation and verification of the proposed technique has been done with CloudSim [16].

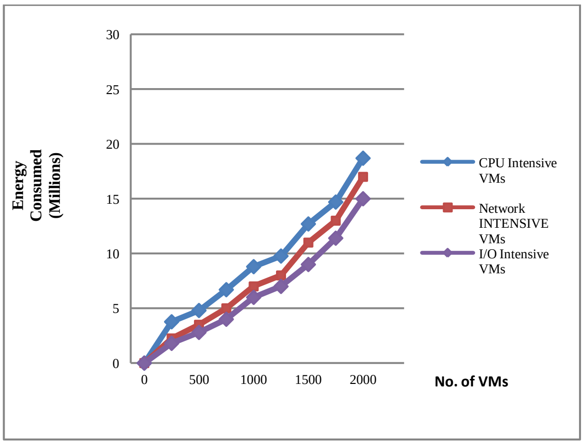

Figure 3. Comparison between CPU Intensive, Network Intensive and I/O Intensive VMs

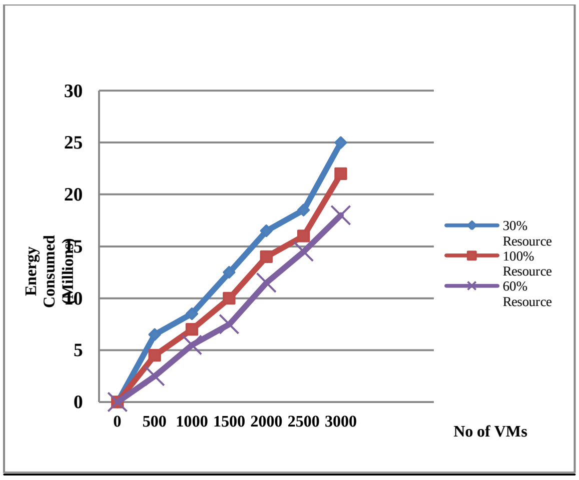

Figure 4. Comparison between 30%, 60% and 100 % overall resource usage

6. CONCLUSION

In this proposed technique, we schedule the VMs using the optimal energy efficient algorithm. This technique finds an operational optimal point which reduces the energy consumption for a desired value of resource utilization**.** The operational optimal point depends and varies on the number of tasks, type of tasks and resource availability. In this algorithm we classify the VMs and schedule them in such way to balance all the types of resources so that the energy consumption is minimized and resources utilization is maximized.

We executed the algorithm and schedule the VMs on different % of resources and concluded at the 60% of overall resources usage is the operational optimal point at which the energy consumption is minimal. This technique optimizes resource utilization and energy consumption is reduced which leads to green environment.

7. REFERENCES

[1] EDS and NCC, The Green IT paradox: Results of the NCC Rapid Survey, EDS; NCC, 2009.

[2] Green Grid 2010. Unused Servers Survey Results Analysis. Green Grid report.

[3] Google Inc. The Big Picture FAQs - Google Green. http://www.google.com/intl/en/green/bigpicture/ references.html, 2013- 07-11.

[4] Susane Albers (2010),"Energy efficient algorithms", Communication of ACM, vol. 53 No. 5, 86-96.

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

[6] Anton Beloglazov and Rajkumar Buyya. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science, MGC ’10, pages 4:1–4:6, New York, NY, USA, 2010, ACM.

[7] J. Stoess and C. Lang, “Energy management for hypervisor based virtual machines,” *on Proceedings of the USENIX Annual*, 2007.

[8] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari and Arka A. Bhattacharya(2010),,” Virtual Machine Power Metering and Provisioning”, copyright 2010 ACM.

[9] S. Rivoire, P. Ranganathan, and C. Kozyrakis. A comparison of high-level full-system power models. In HotPower’08: Workshop on Power Aware Computing and Systems, December 2008.

[10] Husain Bohra, Ata E, and Vipin Chaudhary. “VMeter: power modelling for virtualized clouds”, *IEEE International Symposium on Parallel & Distributed Processing,Workshops and Phd Forum (IPDPSW)* April 2010.

[11] James William Smith Ali Khajeh-Hosseini Jonathan Stuart Ward Ian Sommerville, CloudMonitor: Profiling Power Usage”, IEEE Fifth International Conference on Cloud Computing, June 2012.

[12] James W. Smith and Ian Sommerville, “Workload Classification & Software Energy Measurement for Efficient Scheduling on Private Cloud Platforms.

[13] Sobir Bazarbayev and Matti Hiltunen, Kaustubh Joshi, William.H.Sanders, Richard Schlichting,”Content-Based Scheduling of Virtual Machines (VMs) in the Cloud”, 2013 IEEE 33rd International Conference on Distributed Computing Systems.

[14] Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, Rajkumar Buyya,”A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments”,24th IEEE International Conference on Advanced Information Networking and Applications, AINA 2010, Perth, Australia, 20-13 April 2010.

[15] Bilgaiyan, S. Sagnika, S. and Das, M.,” Workflow scheduling in cloud computing environment using Cat Swarm Optimization”,Advance Computing Conference (IACC), 2014 IEEE International.

[16] R.Buyya. Cloud Simulator cloudsim version 2.1, GRIDS Lab, http://code.google.com/p/cloudsim, July 27, 2010.