Macro Modelling Approach in Power Aware Scheduling

Dr. Dileep P.

Head of Department (Electronics), Maharaja's Technological Institute, Thrissur
Email: *drdileep70[at]gmail.com*

Abstract: ***The demand for computational power by scientific, business and web-applications has led to the creation of large-scale data centers consuming enormous amounts of electrical power. Despite the improvements in energy efficiency of the hardware, overall energy consumption continues to grow due to increasing requirements for computing resources. The main challenge faced by cloud data centers is this power consumption by the servers. There is no proper scheduling approach for cloud providers to provide an optimal scheduling of the virtual machines (VM) to keep the power consumption minimum and to keep the Quality of Service (QoS). The proposed power aware scheduler (PASE) is based on the macro-modeling of the servers by a-priori estimation of power consumption. The a-priori estimation gives a power metric which effectively represents the relative power consumption of the servers. The ranking of the servers is based on this power metric. Here scheduling is done based on this model and there is measure for power-performance trade-off to maintain the QoS. The power aware scheduling is done by effectively scheduling the VMs by their live migration in the cloud servers based on the power ranking of the servers and also on a measure based on its CPU load.***

Keywords: Cloud computing, OpenNebula, Power aware scheduling, Power profile

1. Introduction

Power is the constraint that influences every design. It is the most important factor in virtually every architecture. The power crisis in the modern world and the influence of mobile devices in the world has catalyzed the area of power minimization. Cost reduction has also played its part in this. In the modern world data intensive computing is an inevitable part in the IT sector and its effectiveness virtually drives the whole sector. Power is the most important issue in data intensive computing. Modern resource-intensive enterprise and scientific applications create growing demand for high performance computing infrastructures. This has led to the construction of large-scale computing data centers consuming enormous amounts of electrical power. Despite of the improvements in energy efficiency of the hardware, overall energy consumption continues to grow due to increasing requirements for computing resources. Moreover, there are other crucial problems that arise from high power consumption. Due to high power consumption cooling system malfunction can lead to overheating of the resources reducing system reliability and device's lifetime. In addition, higher power consumption by the infrastructure leads to substantial carbon dioxide emissions contributing to the greenhouse effect.

The running of the data centers generally requires large amount of energy, and energy consumption is a critical issue for IT organizations. For example in 2006, data centers consumed about 4.5 billion kWh, equaling roughly 1.5% of the total US electricity consumption, and trends show that power consumption keeps growing at 18% annually [1] [2]. Also in 2006 the cost of energy consumption by IT infrastructures in US was estimated as 4.5 billion dollars and it is likely to double by 2011 [3] . Facts like these lead to wide popularity of energy efficient computing models like cloud computing which is a style of computing where massively scalable IT enabled capabilities are delivered as a service to external customers using Internet technologies [4].

In cloud computing one of the main goals is power minimization by virtualizing the servers. The virtualization of servers has decreased the power consumption of the data centers but not adequately enough. There is no effective algorithm to schedule the virtual machines (VMs) in the cloud servers to minimize the power consumption. An effective framework for power minimizing in cloud data centers can further decrease the power consumption. Cloud computing decreases the power consumption and space required by organizations and individuals by leasing computing power to them that is, with thin clients they can do jobs which actually needs big data centers. Now considering the cloud data centers, even though it reduces the power consumption, the servers in these data centers consume large amount of power for their working and this consumption varies according to the load, that is, based on their utilization. But the servers consume more power indirectly in the forms of their cooling systems and Air Conditioners (ACs). Also due to invariant peak load in the cloud, most of the server power is under utilized [5]. This leads to wastage of power. Also the rated power consumption of servers has gone up by 10 times over the last 10 years [6].

There is no optimal solution to reduce this power consumption of the virtualized cloud data centers. Many attempts [7] [8] [9] are made for energy saving in cloud but are not sufficient enough to save power and to maintain the QoS. In the existing data centers CPUs are responsible for approximately half of the energy consumed by the servers [7] [10] [11]. A promising technique for saving CPU energy consumption is dynamic speed scaling [7], in which the speed at which the processor is run is adjusted based on demand and performance constraints. The non-linear dependence between the energy consumption and the performance as well as the high variability in the energy prices result in a non trivial resource allocation. Many other attempts based on the method of Dynamic Voltage Frequency Scaling (DVFS) has been tried in [8] [9]. But the power saved by these methods is marginal compared to total power consumption. They are not efficient enough to decrease the power consumption to a level compared to the total power consumption. In the above approaches, the nodes are either in sleep mode or their voltage-frequency ratio is adjusted to minimize power. These cannot completely minimize the power consumed. The nodes still consumes considerable amount of power. A system does not consume power only when it is in the shutdown state [12] [13]. Other ways cannot decrease the power consumption to this level. The methods like changing the system to sleep mode when not used still consumes about 40-50% of power [12]. In this paper a novel method is proposed based on a power based ranking system of the cloud servers by the a-priori estimation of their power consumption for a power aware scheduling in cloud data centers

2. Previous Work

An energy efficient resource management strategy in virtualized cloud data centers is discussed in paper [3]. The paper also stresses on the importance of power minimization and energy-performance trade-off in cloud. It also presents a decentralized architecture of the resource management system for cloud data centers. It also propose the development of the policies of VM placement such as optimization over multiple system resources, network optimization and thermal optimization. The work is done by a simulation of the cloud environment and power consumption of different policies has been obtained by the simulation.

The paper [8] discuss the method of power aware scheduling of VMs in DVFS enabled clusters. It stresses the need for power minimization and its consequences. This paper focuses on scheduling VMs in a cluster to reduce power consumption via the technique of DVFS. Specifically it is for the design and implementation of an energy efficient scheduling algorithm to allocate VMs in a DVFS enabled cluster by dynamically scaling the supplied voltages. The algorithm is studied via simulation and implementation in a multicore cluster. Here the test results and performance discussion justify the design and implementation of the scheduling algorithm. A discussion about power proportional clusters is done in paper [12]. It gives a detailed study about energy consumption in clusters and about the implementation of a cluster manager which decreases the power consumption. It also explains the power proportionality of different components to the cluster power consumption. The paper also discuss about a cluster manager that schedules load and turns on and off the machines to create a power proportional cluster out of non-power proportional nodes. It also reveals the facts of the average CPU utilization of the servers at data centers like the one of Google that the average utilization is about 30% and most servers operate between 10% and 30% of total capacity. The cluster manager proposed by this paper is for HTTP workloads and is evaluated using both synthetic and real workload traces.

A power saving strategy for grids is introduced in [13]. In light of recently stirred energy consumption concerns, the paper investigates the opportunities for power consumption reduction in Grids. Considering real life Grid traces, it states that there is considerable fluctuations in load. The paper puts forward a peak load dimensioning strategy to derive how many servers need to be installed in computational Grids. In lower loaded periods, there is a potential to save energy by dynamically powering on/off servers to address the actual demand for computational capacity. An appropriate scheduling and power-saving scheme can, under such lower-load conditions, considerably reduce energy consumption. The price paid is that some jobs are executed at sites more remote than closer powered-down ones. Yet, the resulting penalty in consumed bandwidth is rather limited and is expected not to cancel the power consumption advantage. The simulated results are given in the paper.

The paper [1] discuss about EnaCloud, an energy saving application live placement approach for cloud computing environments. With the increasing prevalence of large scale cloud computing environments, how to place the requested applications into available computing servers regarding to energy consumption has become an essential research problem. But existing application placement approaches are still not effective for live applications with dynamic characters. This paper proposes a novel approach named EnaCloud, which enables application live placement dynamically with consideration of energy efficiency in a cloud platform. In EnaCloud, a VM is used to encapsulate the application, which supports applications scheduling and live migration to minimize the number of running machines, so as to save energy. Specially, the application placement is abstracted as a bin packing problem, and an energy-aware heuristic algorithm is proposed to get an appropriate solution. In addition, an over-provision approach is presented to deal with the varying resource demands of applications. This approach has been successfully implemented as useful components and fundamental services in the iVIC platform. Finally, evaluation of the approach is done by comprehensive experiments based on VM monitor Xen and the results show that it is feasible.

The paper [14] proposes a system for energy efficient computing in virtualized environments. It puts forward vGreen, a multi-tired software system for energy efficient computing in virtualized environments. The system comprises of novel hierarchical metrics that capture power and performance characteristics of virtual and physical machines. It also defines policies, which use the metrics for energy efficient VM scheduling across the whole deployment. Results show that vGreen can improve system level energy savings by 20% and 15% across benchmarks and varying characteristics. The key idea behind vGreen is linking workload characterization to VM scheduling decisions to achieve better performance, energy efficiency and power balance in the system. The paper [15] discuss about exploiting heterogeneity in Grid computing for energy-efficient resource allocation. The paper proposes a meta-scheduling algorithm HAMA which exploits the heterogeneous nature of Grid to achieve reduction in energy consumption. The reduction in energy is done by effectively distributing compute-intensive parallel applications on Grid.

The paper [16] proposes a new architecture Green-Cloud for green data center. This architecture enables comprehensive online monitoring, live VM migration, and VM placement optimization.

3. Power Aware Scheduling

The power aware scheduling minimizes the power utilization in the cloud. The scheduler runs in parallel with the default scheduler in OpenNebula. The scheduler minimizes power consumption by decreasing the number of on hosts in the cloud. It also focuses on load balancing for improving the performance. The scheduler uses only the low power nodes to deploy the requested VM. The power aware scheduling is done in two steps namely, 1) Creating a power profile for the cluster nodes and 2) Scheduling the VMs according to this power profile.

A. Power Profiling of Cluster Nodes

The first part of implementing the power aware scheduler is the modeling of the servers in the cloud data centers, that is, creating a power profile for the cluster nodes. A macro-modeling of the servers is done. The macro-modeling is done based on creating a power metric by the a-priori estimation of the power consumption of the servers. Since the power consumption of the servers are dynamic in nature, that is, they vary according to the load so does the CPU utilization. The calculation of dynamic power is not easy and needs constant monitoring and analysis of the power consumption of the servers in the cluster under varying load or an in-built power sensor is required in the hardware.

In the cloud there is no need for computing the total power consumption of the nodes a-priori. Only the power metric for ranking of the physical nodes in the clusters is needed. The common factors of the server power consumption like chassis power are eliminated. So the CPU power consumption is taken as the power metric by taking account to the following facts which were observed in the study done by Intel [17], namely, 1) Disk and memory I/O does not affect server power consumption and 2) CPU utilization linearly affects power consumption.

On average the CPU utilization in cluster servers is assumed to be a maximum of 80%. Studies show that on average this value is about 50%, that is, due to this under utilization, most of the server computing power is wasted. The CPU power consumption is calculated based on values that are available from the data sheet such as Thermal Design Power (TDP) and Idle power [18]. Other values such as leakage power and dynamic power are also used. The leakage power is calculated as 10% of TDP.

Dynamic power is calculated by using the formula,

$$P_d = CV^2f$$

The values of f, the frequency of operation of the processor, V, the working voltage of the processor and C, the capacitance value of the processor is available from the data sheet. The power metric is calculated by the formula,

$$P = (P_{tdp}+P_d+P_{idle})/3 + P_l$$

Where Ptdp is the Thermal Design Power of the processor, Pd is the dynamic power of the processor, and Pidle is the idle power of the processor and Pl is the leakage power of the processor.

This metric gives an approximate value of the relative power consumption of the servers in the cluster. Dynamic power is calculated on the assumption that on average the CPU is utilized 80%. The power metric for each physical node in the cluster is saved in a file along with the node's host id. Using this metric the relative power consumption of the servers in the cloud cluster can be modeled effectively with less effort and cost. A modeling based on the relative values of server power consumption is very effective in this circumstance because the need here is to differentiate the servers based on their power consumption. This metric gives the clear cut distinction between the power consumption among the servers in the cluster.

B. PASE: The Power Aware Scheduler

The power aware scheduler works along with the Open-Nebula scheduler and schedules the virtual machines in the cloud such that the energy consumption is minimized. The power utilization in the cloud depends on three factors, namely,1) The power consumption of each cluster node 2) The number of cluster nodes in the cloud and 3) The work load of each cluster node in the cloud. The power aware scheduler consider the factors above, and minimize the power utilization in the cloud. The scheduler minimize the number of cluster node used, the remaining are suspended (the cluster nodes in the cloud are shutdown). The scheduler uses the cluster nodes with minimum power consumption as described by the power ranking. The VMs are scheduled such that only minimum nodes work and the working nodes consume minimum amount of power compared to other nodes. It also balances the load in the cluster nodes to optimize utilization of the nodes in the on state. The architecture of the scheduler is given in figure 1. The working of the scheduler is discussed in algorithm 5.

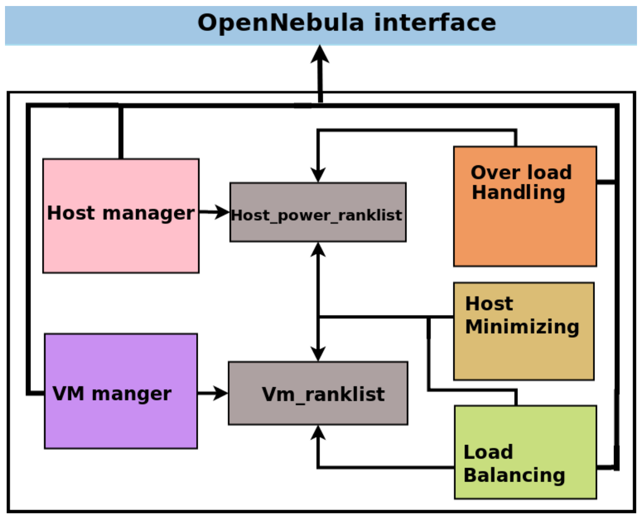

Figure 1: PASE Architecture

The power aware scheduler has five phases.

1) Host management: This host management module retrieves the information of cluster node using OpenNebula interface. The information of resources in each cluster nodes are retrieved by this module, and a database is created. The information retrieved is extracted by XML parsing. The information is used to create a rank list called Host Power Ranklist. The Host Power Ranklist contains the list of cluster nodes

according to the increasing power consumption. This rank list is used by other modules of the scheduler and is also used for the power based ranking of the nodes.

2) VM management: This VM management module retrieves the information of VMs in the cloud running in different cluster nodes. The resources used by each VM is recorded. Here mainly, memory is considered as the resource. This module also creates a rank list called Vm Resource Ranklist. The module uses OpenNebula interface to retrieve the VM details and a database of the VMs is created. The Vm Resource Ranklist is used for load balancing among the nodes.

3) Over Load Handling: Overload handling checks whether the number of not over loaded hosts is below a predefined value. If such a situation is identified, then new hosts which are disabled is changed to enable, these enabled hosts must be lower in power consumption, that is they must have high power ranking. A fixed number of cluster nodes must be in the not over loaded state for satisfying the requirement of customers and attain QoS. It can be denoted by #Requirednotoverloadhost. Here ucpu, fcpu represents used CPU and free CPU respectively. This module works as described in algorithm 1.

Algorithm 1 Overload Handling

begin
1) Notoverload=0
2) For each enabled host
3) Check if the $\frac{fcpu}{ucpu}$ of cluster node $\leq$ lowload then
4) Notoverload+=1
5) If Notoverload $<$ #Requirednotoverloadehost then
6) Enable the first $(\#Requirednotoverloadedhost - Notoverload)$ nodes from Host_Power_Ranklist

end

Algorithm 2 Pseudo code for Host minimizing

begin
1) Nooflowload=0
2) For each enabled host in the cloud
3) Check if $\frac{fcpu}{ucpu}$ $<$ lowload then
4) Nooflowload+=1
5) If Nooflowload $>$ #maxlowloadhost then
6) Disable one cluster node as in algorithm 3.

end

Algorithm 3 Pseudo code for Cluster node disabling

begin
1) For each enabled host in the Host_Power_ranklist.
2) Sort the VM in the host according to the increasing rank in the Vm_Ranklist.
3) For each VM, check if it can be migrated to low load host other than in which it is currently running then
4) record it.
5) If all VMs in the cluster node can migrate then
6) Migrate all VM using the recorded details.
7) Disable the cluster node.

end

Algorithm 4 Pseudo code for Load Balancing

begin
1) Create the list of over load host and low load host .
2) If overload host list or low load host list is empty then stop
3) Sort the cluster nodes according to decreasing load.
4) Take the first and last cluster node from the sorted list.
5) If first and last are same then stop
6) Take a VM from the first cluster node which is not already migrated
7) If no such VM exist then take first as next cluster node in the list and go to step 5.
8) Repeat the above steps .

end

4) Host Minimizing: The host minimizing module minimizes the number of cluster nodes used in the cloud. It checks for a situation in which large number of cluster nodes are in low load. The upper bound of the low load cluster node is taken as #maxlowloadhost. If the number of low load host is greater than the #maxlowloadhost , then this module minimizes the number of host by disabling some cluster nodes according to their power ranking. It disables the cluster nodes according to their decreasing power. The highest power consuming cluster node is disabled first. The live migration of VMs is used here. The VM from the disabling host migrates to the other enabled cluster node. The host minimizing module only try to disable one host at a time, so the scheduler does not decrease the QoS. If more number of cluster nodes are tried to be minimized, it needs a lot of migration and this slows down the performance and thus decreases the QoS, that is, user requirement. The working of this module is described in algorithm 2. The host minimizing module disables only one cluster node at a time. But repeated cycle of scheduler minimize the number of nodes very efficiently. Also this minimization does not significantly affect the performance of the cloud.

Algorithm 5 Working of PASE

begin
1) Initially disable all the hosts.
2) Enable only minimum number of cluster nodes that have lowest power.
3) Repeat
4) Handle the overload condition as given above.
5) Try to minimize the number of hosts.
6) Try to balance the load.

end

5) Load balancing: Load balancing module balances the load in the cluster nodes by migrating VMs from overloaded host to low load host. The load balancing migrates VM only once in a cycle. This also ensure the QoS. This does not seriously affect the performance of the cloud. The working of this module is discussed in algorithm 4.

The power aware scheduler suspends (shutdown) every disabled host. This minimizes the power utilization in the cloud. These hosts are turned on through LAN. Wakeonlan is used for this purpose. Syntax is wakeonlan <Ipaddress>. In this cloud the IP address is the name of the physical node itself.

4. Experimental Validation and Results

This section discuss the results obtained by implementing the framework for power aware scheduling in cloud. Open-Nebula was used for creating the cloud environment [19]. KVM was used as the hypervisor in the cluster nodes [20]. Linux operating system was used for obtaining the results. The framework was done using C++. For implementing the cloud infrastructure two laptops with Intel(R) Core(TM) i3 CPU M 350 2.27GHz processor having the Virtualization Technology (VT) support was used. This can be found by checking for the vmx flag in cpuinfo file. The cloud infrastructure was developed first using OpenNebula and it was installed at the front-end node. Then the back-end nodes were configured after installing KVM in them and bridges and virtual networks were created. The administrator at the front-end controls the whole infrastructure. There are also local administrators at the cluster nodes. Password less SSH access to the nodes were enabled. Also the image repository of the front-end was shared with all the other nodes. The power aware scheduler starts along with the OpenNebula one scheduler. The results obtained are discussed below. This section also verifies how the scheduler works in different conditions.

A. Initialization of scheduler

Initially the scheduler enables the hosts which have the highest power ranking. Here 192.168.2.32 has lowest power consumption that is, highest power ranking, so it is enabled. 192.168.2.31 with high power consumption is disabled.
oneadmin@dileep-laptop: onehost list

ID	*NAME*	*RVM*	*TCPU*	*FCPU*	*ACPU*	*TMEM*	*FMEM*	*STAT*
3	192.168.2.32	0	400	398	398	3805700	2750212	On
6	192.168.2.31	0	400	400	400	0	0	off

B. Overload handling

An event for overload handling is given below. There is only one host which is enabled. The amount of free CPU in this host is very low in the second result of onehost list. In this situation it can be seen that the over load handling module in the scheduler enabled the next host, here it is 192.168.2.31.
oneadmin@dileep-laptop: onehost list

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	1	400	379	379	3805700	2750212	On
6	192.168.2.31	0	400	400	400	0	0	off

The host with ID 3 is initially on and ID 6 is off. This situation is shown above. After some time the host with ID 3 becomes overloaded. To handle the overloaded condition, the host with ID 6 is also turned on by the scheduler as depicted below.
oneadmin@dileep-laptop: onehost list

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	1	400	254	254	3805700	2750212	On
6	192.168.2.31	0	400	400	400	3677868	3030333	On

C. Host minimization

The host minimization module of scheduler tries to minimize the number of enabled hosts. An event is given below.
oneadmin@dileep-laptop: onehost list

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	1	400	158	158	3805700	2750212	On
6	192.168.2.31	1	400	400	400	3678990	2334453	On

In the situation above, the host with ID 3 is overloaded and ID 6 is low loaded.
oneadmin@dileep-laptop: onehost list

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	2	400	373	373	3805700	2750212	On
6	192.168.2.31	1	400	392	392	3678990	2334453	On

Gradually both the hosts with ID 3 and ID 6 becomes low loaded as shown above.

In the above example it can be seen that the load on both cluster nodes decreased(in second onehost list). In this situation the highest power host is disabled by migrating the VM running on this host to 192.168.2.32.
oneadmin@dileep-laptop: onehost list

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	2	400	254	254	3805700	2750212	On
6	192.168.2.31	0	400	400	400	3678990	2334453	Off

D. Load Minimization

The load minimization tries to minimize the load in the host. An event is given below. 192.168.2.31 is low loaded and 192.168.2.32 is over loaded. In this condition one VM in the over loaded host is migrated to the second one, which is low loaded.

oneadmin@dileep-laptop: onehost list

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	2	400	254	254	3805700	2750212	On
6	192.168.2.31	0	400	400	400	3678990	2334453	On

The state of the hosts during the migration state is shown below.

oneadmin@dileep-laptop: onehost list

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	2	400	252	252	3805700	2750212	On
6	192.168.2.31	1	400	400	400	3678990	2334453	On

oneadmin@dileep-laptop: onehost lis

ID	NAME	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
3	192.168.2.32	1	400	312	312	3805700	2750212	On
6	192.168.2.31	1	400	357	357	3678990	2334453	On

It can be seen that after the load minimization, the load on both nodes are balanced.

5. Conclusions & Future Scopes

This report proposes a novel framework for a power aware scheduling in cloud. The scheduling is based on a macromolecular of the cluster nodes in the cloud data center based on the a-priori estimation of their power consumption. A power metric for ranking the cluster nodes in the cloud is calculated based on this a-priori estimation. This ranking model is used for determining the ranks of the hosts for the novel power aware scheduling of VMs in the cloud. In this work a cloud infrastructure was implemented using OpenNebula, an open source framework. The power minimization is achieved by scheduling the VMs among the nodes such that minimum number of nodes are working. The scheduler also implements load balancing for improving performance. It also eliminates hot-spots by reducing the workload and thus increases the longevity of the machines.

Some future enhancements that can be made on the framework discussed so far are presented below. The proposed macro model of the cloud servers can be extended to a dynamic power based modeling. Improving the power-performance trade-off is a future prospect. The focus is towards developing an efficient, decentralized and scalable power aware algorithms for resource allocation.

References

[1] Bo Li, Jianxin Li, Jinpeng Huai, Tianyu Wo, Qin Li, and Liang Zhong, "Enacloud: An energy-saving application live placement approach for cloud computing environments," in Proc. of IEEE International Conference on Cloud Computing, 2009.

[2] Report to Congress on Server and U.S. Environmental Protection Agency ENERGY STAR Program Data Center Energy Efficiency, "http://www.energystar.gov/ia/prod_development/downloads" .

[3] Anton Beloglazov and Rajkumar Buyya, "Energy efficient resource management in virtualized cloud data centers," in Doctoral Symposium, Proceedings of the 10th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2010), Melbourne, Australia, May 2010.

[4] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities," in In proceedings of the 10th IEEE International conference on High Performance Computing and Communications (HPCC'08), 2008.

[5] L. A. Barroso, U. H lzle, "The case for energy-proportional computing," Computer, vol. 40, no. 12, pp. 33–37, 2007

[6] R. Raghavendra, B P. Ranganathan, V. Talwar, Z. Wang, and X.Zhu, "No power struggles: Coordinated multi-level power management for the data center," in In Thirteenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS08), March 2008.

[7] Rade Stanojevic and Robert Shorten, "Distributed dynamic speed scaling," in Proceedings of the 29th conference on Information communications, January 2010, pp. 426–430.

[8] Gregor von Laszewski, Lizhe Wang, Andrew J. Younge, and Xi He, "Power-aware scheduling of virtual machines in dvfs-enabled clusters," in Proceedings of the 2009 IEEE International Conference on Cluster Computing, August 31 - September 4 2009.

[9] G.Magklis, G. Semeraro, D. Albonesi, S. Dropsho, S. Dwarakadas, and M. Scott, "Dynamic frequency and voltage scaling for a multiple-clockdomain microprocessor," in IEEE Micro, vol. 23, no. 6, 2003, pp. 62–68.

[10] A. Greenberg, J. Hamilton, D .A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," ACM Computer Communications Review, vol. 39, no. 1, 2009.

[11] W. Hammond, "Efficient power consumption in the modern datacenter. available online:

www.research.ibm.com/aceed/2005/proceedings/hammod.ppt,"
[12] Sara Alspaugh, Laura Keys, and Andrew Krioukov, "Power proportional cluster," in University of California, Berkeley, March 2010
[13] C. Develder, M. Pickavet, B. Dhoedt, and P. Demeeste, "A power-saving strategy for grids," in 2nd Int. Conf. on Networks for Grid Applications, Beijing, China, October 2008
[14] Dhiman G., Marchetti G., and Rosing T.S., "vgreen: A system for energy efficient computing in virtualized environments," in In Proceedings of the 14th IEEE/ACM International Symposium on Low Power Electronics and Design, 2009
[15] Saurabh kumar Garg and Rajkumar Buyya, "Exploiting heterogeneity in grid computing for energy-efficient resource allocation," in Proceedings of the 17th International Conference on Advanced Computing and Communications (ADCOM 2009), Bengaluru, India, December 2009
[16] Liang Liu, Hao Wang, Xue Liu, Xing Jin, WenBo He, QingBo Wang, and Ying Chen, "Greencloud: a new architecture for green data center," in International Conference on Autonomic Computing, Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session, June 2009, pp. 29–38
[17] Tridib Mukherjee, Georgios Varsamopoulos, Sandeep K. S. Gupta, and Sanjay Rungta, "Measurement-based power profiling of data center equipment," in Proceedings of the 2007 IEEE International Conference on Cluster Computing, 2007, pp. 476–477.
[18] Processor data sheets, "http://www.intel.com,".
[19] OpenNebula, "http://www.opennebula.org,".
[20] KVM,http://www.linux-kvm.org,

Author Profile

Dr. Dileep. P. is presently working as Head of Department (Electronics) at Maharaja's Technological Institute, Thrissur. He has completed his PhD from Anna University, Chennai in 2013. He took his B.Tech in Electronics Engineering with First Class from Cochin University of Technology, Kerala in the year 1998 and ME in Embedded System Technologies with First Class from Vinayaka Mission's University, Salem in the year 2007. He has an experience of more than 25 years in teaching and administration in various Govt. Polytechnic Coilleges in Kerala. He has published a number of papers in National and International Journals. In addition to that he has presented a paper on Nanotechnology at NESCOT, London in 2015 in connection with the UK-India Education and Research Initiative (**UKIERI**) project.