Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2018, 5(1): 61-68

Research Article ISSN: 2394 - 658X

Energy Efficient DVFS with VM Migration

Nishant Kumar, Dr. Raj Kumar and Dr. Mayank Aggrawal

Department of Computer Science & Engineering, Gurukula Kangri University, Haridwar, India
nishant@gkv.ac.in

ABSTRACT

Cloud Computing offers efficient computing with Pay-as-you-go models. It is now easy for consumer to start without need of initial setup, which saves a lot of infrastructure cost. As consumers are subscribing to the cloud, the load is increasing on the data centers, thus data centers are in need for more resources and more power. And all this process is increasing the carbon footprint and polluting environment. Now the time has come when we require efficiency in term of power. We really need to look for mechanism how the power can me be managed to be more efficient. This paper suggests the Green Architecture Framework and also suggests to use of Dynamic Voltage Frequency Scaling (DVFS) as per the load requirement which results in better energy efficiency

Keywords: Power Efficiency, Energy Efficiency, Green Cloud, DVFS, VM Migration

INTRODUCTION

Cloud computing is to provide the resources as per need and at any place and at any time with the resources pooled up at Data centres. Data centres uses virtualization to provide complete utilization of servers [1]. Cloud computing delivers an infrastructure (IAAS), platform (PAAS) and software (SAAS) as services to the consumers on pay-per-use model. All the big giants like Google, IBM, Microsoft and Yahoo are deploying their data centres to provide the cloud computing services.

In last few years' research has been done in various trades of Cloud Computing SAAS to make it highly efficient technology. Now when big IT companies has started showing their trust and started putting investment to provide services to their clients in the cloud infrastructure, research dimensions have changed towards sustaining, reliability, security, etc. Cloud community has grown big enough, its impact in climate and environment has become a concern in terms of carbon footprint. To achieve lower power consumption without comprising the efficiency is a major task. [2] has shown that DVFS algorithm which is quite effective solution to perform in Cloud Environment. This technology is the best in itself as it is dynamic in nature and controls the power flow in the host or computer system as per the current load which means less work then less power and as the workload increases power frequency increases as well. In this paper we have defined an architectural framework for efficient cloud computing and proposed a method to achieve high efficiency in terms of power, by combining DVFS with VM Migrations.

In order to get the efficiency at hardware level one thing which works really well is Dynamic Voltage Frequency Scaling (DVFS). A technique for minimizing the power consumption by gearing up and down the voltage frequency as per the requirement is discussed in [2]. This way energy could be saved dynamically. The other way to minimize the power consumption is ON/OFF method, this works with the threshold value. If CPU is underutilized, then all process running will be migrated to the other host and switched off to save power. When the host is over-utilized then another host will be switched on to manage the load. But migrating the VM to other host also need energy, it is also an important point to be taken into account [3-4].

Pinheiro and Bianchini *et al* [5] states that the main technique to minimize the power consumption is by having minimum hosts with full load &by turning off idle machines, but this technique lacks in power and performance trade-off. The algorithm periodically checks the usage of resources to minimize power consumption by switching them off if not required. In this technique Load balancing is dependent on single application which leads to the single point power failure (SPF). Chase *et al* [6] states that main task is to find the requirement of resources by the application at the current load. It has suggested an economic bidding model. This enables negotiations of SLA's as per the budget and current QoS, thus it proposes gain by maintaining the usage cost of resources and the benefits comes from full utilization of resources. In this energy is reduced switching the system ON and OFF as per the requirement. Kusic *et al* [7] used Limited Look Ahead Control for minimizing the power consumption in heterogeneous virtual systems. Kalman Filter was used to reduce power consumptions along with minimum SLA violations by predicting the future requests.

Live migration of VMs was automated Forsman *et al* [8]. It works on two strategies, push and pull. In push strategy overloaded physical machines attempt to give their load to less utilized physical machines. In pull strategy, the physical machines which are underutilized, request the workload from over utilized nodes. The paper focuses on balancing the workload efficiently and re-distributes it as per load. Re-distribution of the workload considers three factors, physical machines state after migration, cost of migration and expected workload. Song *et al* [9] used virtualization technology to use cloud data-centre resources on the run-time and to make use of servers efficiently for green computing. It proved that the Variable Item Size Bin Packing (VISBP) algorithm has better performance in hot spots migration as compared to the then existing algorithms. Han *et al* [10] used the concept of putting virtual machine to few hosts and switching idle machines of. Though large number of consolidations of VMs may result in high SLA violations. Wang *et al* [11] proposed a task scheduling method which dispatches the tasks to the active or running servers by using servers as less as possible and focused on adjusting the execution frequency of processors of relative cores to save energy. It also proposed a processor-level migration algorithm to re-schedule the remaining task among processors on an individual server. Wajid *et al* [12] shows concern about environment by focusing on the CO2 and suggested an eco-approach, it involved mainly three phases Measure, Create and Test. The first phase quantifies the energy consumption and environmental impact, second develops techniques and softwares to reduce energy consumption and CO2 emissions of cloud applications and third tests then outcomes of the above two phases. Prathibha *et al* [13] proposed the system to minimize the energy consumption in the cloud Data Centres by sending the tasks to the energy efficient data center and also running the virtual machine at required frequency using enhanced weighted DVFS mechanism. To further reduce the energy consumption due to networking devices such as switches and routers, a DENS based scheduler with hybrid load balancer is proposed in this work.

GREEN CLOUD EFFICIENT ARCHITECTURE

Next generation of cloud is to create a network of virtual servers or services (database, application, hardware etc.) so that users can be benefited with the rapid deployment and development from anywhere on demand at a very competitive price. Fig. 1 shows the architecture for energy efficient cloud. [14] has proposed the above architecture for energy efficient cloud, in that architecture we have added Pattern / Behaviour Monitor and Dynamic Power Management.

Consumers/Brokers

There are many users of cloud who is using cloud and getting their service request from around the world. It is important to understand the type of users and their terminology, cloud consumers are the users of cloud in the form of broker and end user, consumer can be a company who deployed application to serve their customers or it can be an end user who is using cloud to speed-up intensive simulations.

Energy Efficient Service Allocator

Allocation of cloud service to its consumer requires to be energy efficient in-order to achieve overall datacentre efficiency. It need to check on few steps to manage the resources efficiently.

Negotiator/Arbitrator

Cloud service quality measurement scheme is known as SLA. Negotiation in SLA between cloud provider and consumer needs to be done prior using service. In this cloud provider commits the service quality for its consumer, and decided the penalty in case of not providing the committed quality of service. In case of web-applications, QoS metric is that 95% request are to be served in 3s.

Service Analyser

Service analyser keeps monitoring to gather information of load and energy from VM manager. This helps in accommodating the current service request.

Exclusive Privileges

There are consumers who have asked for high SLA and using services heavily can be consider as important users, this grading helps in granting special concern, service and prioritizing over other user and consumers.

Energy Monitor

Its gives the information of the use of resources by virtual machines and physical machines to the VM manager to take efficient allocation decision.

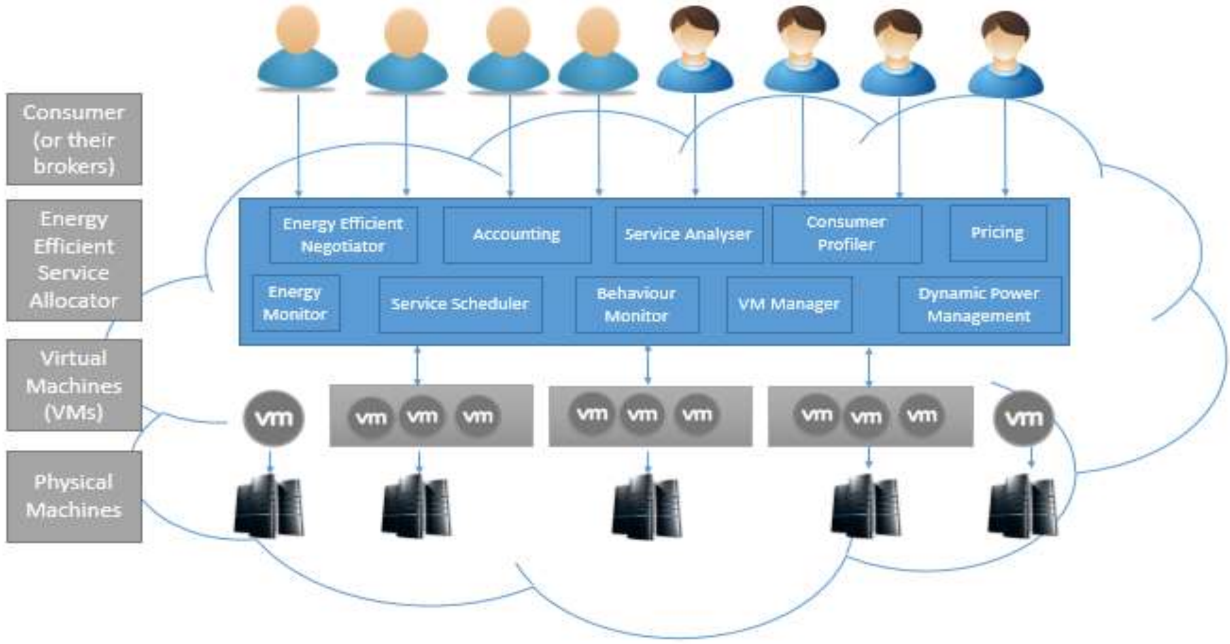

Fig. 1 System Architecture

Service Scheduler

It assigns the service requests to virtual machines and determines the resource entitlements for the allocated virtual machines from Resource Accounting. If the customer asked for auto-scaling as per the load requirement, this is also managed by scheduler when to increase the servers to accommodate the requests.

Pattern/Behaviour Monitor

Monitoring application behaviour can help in efficient allocation. Application are being used by many users and it is using resources accordingly. This might be possible that application won't be using maximum resources all the time. For any application to run on efficiently we need to look its historical data to learn the pattern of usages. Historical data is further being processed by calculating mode and median of the usages by application, to find the near maximum usage of the application and store it in a variable. This variable further can be used to calculate availability of the resources after assumption of near maximum usage. This value will be an input for VM Manager and will help in making trade-off between performance and SLA Violation. This variable can minimize migrations of virtual machines.

VM Manager

VM manager keeps track the use of the virtual machine and their resource usage. It is responsible to start the new virtual machines on a physical machine and also re-allocating virtual machine to another less used physical machine as per the load requirement.

Accounting

It takes information of resources cost consumed by virtual machine to complete the tasks.

Dynamic Power Management

In traditional approach VM migrations are used for energy efficiency but unnecessary VM migrations results in excess power consumption, also putting the system down and turning up also wastes lot of energy. We have proposed to use VM Migrations with algorithms like Dynamic Voltage Frequency Scaling(DVFS) this results in lesser number of migrations and also lesser shutdowns of systems.

VMs

Evolution of Virtual Machines enabled users to run multiple independent machines on the same hardware at the same time with different or same operating system environment. These machines can be dynamically start and stop as per the load requirement. Hence it provides the high flexibility of using machines. Multiple virtual machines can concurrently run different or same applications based on different operating system on a single physical machines. It also provides the capability to run on any architecture and can also migrate to another physical machine on run-time. With this capability managing physical machine load is easy and processor can be managed to run at lower speeds to save energy.

Physical Machines

Physical machines are the underlying infrastructure for creating virtualized services to meet the processing demands.

ENERGY EFFICIENT ALGORITHMS

Virtualization is the concept which has changed the picture of Data Centres by enabling them to use the resource to the maximum. Virtualization has several properties like resizing the virtual machine when it is not being used completely which can be expand later as usages goes up, one host can have multiple virtual machines as per the configuration. In [3] live migration another important aspect of virtualization enables transfers of virtual machine from one system to another without letting down the server in case the host is less utilized and idle nodes now can be switched to sleep mode to save power. Currently, resource allocation is based on performance as written in SLAs (Service Level Agreement), not focused on energy saving. Now as the technology is maturing time has come to trade-off between performance and energy efficiency as energy is one major aspect which leads to the cost of power and also affects the environment. Making system energy efficient leads to some QOS issues and does also affects the SLA at one point. Now it is required to think the trade-off between performance, energy efficiency and SLA. To save power it is must to fully utilize all the system in dynamic manner. When the request of new VM comes, there are three important things to be considered VM Allocation, VM Selection Policy and Power Policy. This paper proposed to use all these algorithms together to get the best efficiency while maintaining the SLA, to its best.

VM Allocation Policy

VM Allocation policy examines the nature of request and decides where to put it efficiently weather new host are needed to boot-up or any running host can be utilized which has sufficient free resources. Allocation of policy conducts in two parts. In first, request is coming for virtual machine provisioning and turning on VM on hosts. Second part handles the optimization of virtual machine allocation. First one is similar to bin packaging problem with variable bin sizes and prices. To solve the problem Best Fit Decreasing (BFD) algorithm is applied [31].

In Modified Best Fit Decreasing algorithm, sorts all the virtual machine in the decreasing order of their utilization and allocate each virtual machine to hosts which accommodate the new virtual machine with the least increase in power usage. This allows leveraging the heterogeneity of resources by choosing the most power-efficient nodes first. The pseudo-code for the algorithm is presented in Algorithm 1. The complexity of the Allocation part of the algorithm is n · m, where n is the number of VMs that have to be allocated and m is the number of hosts.

Algorithm 1 Pseudocode MBFD [12]
Input: hostlist, vmlist
Output: Allocation of Virtual Machines

```
Sortvmlist.DecreasingUtilization()
foreach vm in vmlist do
        minpower <- MAX
        allocatedHost <- NULL
        foreach host in hostList do
                if host has sufficient for the load of vm then
                        power <- estimatepower(host,vm)
                        if power < minpower then
                                allocated host <- host
                                minpower<-power
        if allocatedHost !=NULL then
                allocate vm to allocatedHost
return allocation
```

VM Selection Policy

It is to decide when to migrate virtual machine from one host to another. The Migration policy sets two threshold lower and upper. Migration will take place only in the condition if host reaches to lower threshold or if utilization crosses the upper thresholds. In the most efficient algorithm according to [11], MM Policy(The Minimization of Migrations), selects minimum virtual machine which are need to migrate to lower the CPU utilization. Let Vj be a set of VMs currently allocated to the host j. Then P(Vj) is the power set of Vj. The MM policy finds a set R ∈ P(Vj) defined below.

$$R = \left\{ \left|\ s \mid s \in p(Vj), Uj - \sum_{v \in S} Ua(v) \right) < Tu, |S| \to \min \right\}, Vj, \phi \quad \begin{matrix} \textit{if } Uj>Tu \\ \textit{if } Uj<Tl; \\ \textit{otherwise} \end{matrix}$$

where Tu is the upper utilization threshold; Tl is the lower utilization threshold; uj is the current CPU utilization of the host j; and u(v) is the fraction of the CPU utilization allocated to the VM [9].

The problem of over-utilization in MM algorithm is presented in the Algorithm. Initially the algorithm sorts all the virtual machine in the decreasing order of the CPU utilization. It keeps checking the list of virtual machines to find a virtual machine which is best to migrate from the host. The best VM to migrate is the one which satisfies two condition. First, the VM should have the utilization higher than the difference between the host's overall utilization and the upper utilization threshold. Second, if the VM is migrated from the host, the difference between the upper threshold and the new utilization is the minimum across the values provided by all the VMs. If there is no virtual machine matches the upper criteria then algorithm selects the virtual machine with the highest utilization and removes it from the list of VMs, and proceeds to a new iteration. The algorithm stops when the new utilization of the host is below the upper utilization threshold. The complexity of the algorithm is proportional to the product of the number of over- utilized hosts and the number of VMs allocated to these hosts.

Algorithm 2 Pseudocode MM [15]

```
Input: hostlist
Output: migrationlist
foreach host in hostlist do
        vmlist <- host.getvmlist()
        vmlist.sortdevreasingutilization()
        hutil <- host.getutil()
        bestfitutil <- MAX
        while htuil > THRESH_UP do
                foreach vm in vmlist do
                        if vm.getUtil() > hUtil – THRESH_UP then
                                t <- vm.getUtil() – hUtil + THRESH_UP
                                if t < bestfitutil then
                                        bestfirutil <- t
                                        bestfitvm <- vm
                        else
                                if bestfitutil = MAX then
                                        bestfitvm <- vm
                                break
                hutil <- hutil – bestfitvm.getutil()
                migrationlist.add(bestfitvm)
        if hutil < THRESH_LOW then
                migrationlist.add(host.getvmlist())
                vmlist.remove(host.getvmlist())
return migrationlist
```

The most efficient power policy is DVFS [2], which allows system to shut-down only in case of zero utilization, if the host is less utilized it dynamically lowers the frequency. There are few factors which are needed to be considered for power consumption in VM Migration and Shutting Down the system. There is a significant power consumption in migration of VMs similarly shutting down and again booting –up of the host also takes power. The DVFS mechanism helps in less VM migrations, less shutdowns. Lowering the frequency as per the usage save a lot. There are many cases when system is near to less utilization threshold but still taking full energy. At such points DVFS helps to save energy.

PROPOSED MODIFIED DVFS

Currently DVFS does not use any Migration technique, but manages the Energy dynamically by changing the frequency. In this paper we have used DVFS along with migration using modified MM Migration algorithm. This has resulted in better energy efficiency. In the existing MM policy [16], the machine was used to migrate if only it reaches to higher threshold or lower threshold, there is no power reduction in between. Our approach is providing efficiency specifically for every load.

In modified MM algorithm there is no lower threshold only higher threshold is available. When the system is less utilized then instead of lower threshold, DVFS is capable of reducing the energy consumption by lowering the frequency as per requirement. This function save huge amount of energy and time. Proposed mechanism reduces the downtime by not migrating at low utilization level. This not only reduces the downtime but the overhead of shifting VM to other system. In the scenario of cloud, a little downtime leads to violation of SLA thus our model results in lesser number of migrations and lesser SLA Violation and less cost of power.

Modified DVFS

```
publicclass effcom {

	publicstaticvoid main(String[] args) throws IOException {
		boolean enableOutput = true;
		boolean outputToFile = true;
		String inputFolder = Dvfs.class.getClassLoader().getResource("workload/planetlab").getPath();
		String outputFolder = "output";
		String workload = "20160617"; // PlanetLab workload
		String vmAllocationPolicy = "dvfs";
		String vmSelectionPolicy = "mmt";
		String parameter = "0.8";

		new PlanetLabRunner(
				enableOutput,
				outputToFile,
				inputFolder,
				outputFolder,
				workload,
				vmAllocationPolicy,
				vmSelectionPolicy,
				parameter);
	}
}
```

EXPERIMENTAL SETUP

For the results in Cloud computing environment with multiple nodes, simulators provide the real-time environment. The CloudSim toolkit [17] has been chosen as a simulation platform, as it is most preferred and modern simulation framework for Cloud computing environments. In contrast to alternative simulation toolkits (SimGrid, GangSim), CloudSim allows the modelling of virtualized environments, and supports on-demand resource provisioning, and their management. It also enables energy- aware simulations capability. Apart from the energy consumption modelling it has the ability to simulate service applications with dynamic workloads.

We have simulated a data centre that has 800 heterogeneous physical nodes, half of which are HP ProLiant ML110 G4 servers, and half consists of HP ProLiant ML110 G5 servers. The server's CPUs frequency are mapped onto MIPS ratings: 1860 MIPS each core of the HP ProLiant ML110 G5 server, and 2660 MIPS each core of the HP ProLiant ML110 G5 server. Each server is modelled to have 1 GB/s network bandwidth. For our experiments we have used data provided as a part of the CoMon project, a monitoring infrastructure for PlanetLab [18]. The amount of RAM is divided by the number of cores for each VM type: High-CPU Medium Instance (2500 MIPS, 0.85 GB); Extra Large Instance (2000 MIPS, 3.75 GB); Small Instance (1000 MIPS, 1.7 GB); and Micro Instance (500 MIPS, 613 MB). Initially the VMs are allocated according to the resource requirements defined by the VM types. However, during the lifetime, VMs utilize less resources according to the workload data, creating opportunities for dynamic consolidation.

SIMULATION RESULTS AND ANALYSIS

Using the workload data of planetlab 20110303 which has 1052 VM, on over 1000 physical Nodes we have simulated all the proposed host overloading detection algorithm (THR, IQR, MAD, LR and LRR) and a VM Selection Algorithm MMT.

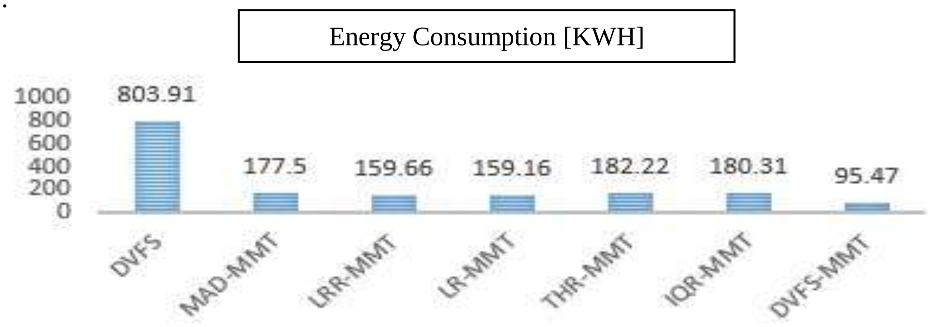

Fig. 2 Energy Consumption Comparison Chart

Table-1 Energy Consumption Comparison Table

Policy	Energy(kWH)	VM Migration	Average SLA Violation(%)	Shutdowns
NPA	3013.50	0	0	657
DVFS	803.91	0	0	657
MAD-MMT	177.50	23733	9.77	5657
LRR-MMT	159.66	26717	9.51	4898
LR-MMT	159.16	26717	9.51	4898
THR-MMT	182.22	24175	9.93	5655
IQR-MMT 1.5	180.31	23601	9.82	5667
DVFS-MMT(Proposed)	95.47	2539	14.34	976

The comparison result clearly indicates that energy consumption has decreased by 60% in comparison with existing best available algorithm though it results in slight increase in SLA violations. We considered that SLA can be compromised in SLA document but the cost of power can't be compromised and power efficiency has greater impact on environment. Reduction in power contributes to reduced carbon footprints.

CONCLUSION

The results clearly show that proposed model with modified algorithms gives better results in terms of energy consumptions. Energy consumption is one of the major problems in cloud Computing thus by using proposed approach we can easily overcome with this issue. Using dynamic voltage frequency scaling with VM migration helps in reducing the energy consumption. The model though results in more SLA violation which can be compromised and can be taken up in Future.

REFERENCES

[1] P Barham, B Dragovic, K Fraser, S Hand, T Harris, A Ho, R Neugebauer, I Pratt and A Warfield, XEN and the Art of Virtualization, *ACM SIGOPS Operating Systems Review*, **2003**, 37(5),164-177.
[2] T Guerout, T Monteil, G Da Costa, R Neves Calheiros, R Buyya and M Alexandru, Energy-Aware Simulation with DVFS, *Simulation Modelling Practice and Theory*, **2013**, 39, 76–91.
[3] C Clark, K Fraser, S Hand, JG Hansen, E Jul, C Limpach, I Pratt and A Warfield, Live Migration of Virtual Machines, *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, **2005**, 273–286.
[4] W Voorsluys, J Broberg, S Venugopal and R Buyya, Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation, *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), **2009**, 5931 LNCS, 254–265.
[5] E Pinheiro and R Bianchini, Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems, *Technical Report DCS–TR*–440, **2001**, DCS–TR–440, 1–8.
[6] JS Chase, DC Anderson, PN Thakar, AM Vahdat and RP Doyle, Managing Energy and Server Resources in Hosting Centres, *Proceedings 18th ACM Symposium Operating System Principles*, **2001**, 103–116.
[7] D Kusic, JO Kephart, JE Hanson, N Kandasamy and G Jiang, Power and Performance Management of Virtualized Computing Environments Via Look Ahead Control, *International Conference on Autonomic Computing*, **2008**, 3–12.
[8] M Forsman, A Glad, L Lundberg and D Ilie, Algorithms for Automated Live Migration of Virtual Machines, *Journal of Systems and Software*, **2015**, 101, 110–126.
[9] W Song, Z Xiao, Q Chen and H Luo, Adaptive Resource Provisioning for the Cloud using Online Bin Packing - Wagner, *IEEE Transactions on Computers*, 2014, 63 (11), 954-967.
[10] G Han, W Que, G Jia and L Shu, An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing, *Sensors* (Switzerland), **2016**, 16(2), 1–17.
[11] S Wang, Z Qian, J Yuan and I You, A DVFS Based Energy-Efficient Tasks Scheduling in a Data Center, *Special Section on Emerging Trends, Issues, and Challenges in Energy-Efficient Cloud Computing*, **2017**, 5, 13090–13102.
[12] U Wajid, C Cappiello, P Plebani, B Pernici, N Mehandjiev, M Vitali, M Gienger, K Kavoussanakis, D Margery, DG Perez and P Sampaio, On Achieving Energy Efficiency and Reducing CO_2 Footprint in Cloud Computing, *IEEE Transactions on Cloud Computing*, **2016**, 4(2), 138–151.
[13] S Prathibha, B Latha and G Sumathi, Improving Energy Efficiency of Computing Servers and Communication Fabric in Cloud Data Centres, *Proceedings of 2nd IEEE International Conference on Research in Computational Intelligence and Communication Networks*, **2017**, 17–21.

[14] R Buyya, A Beloglazov and J Abawajy, *Energy-Efficient Management of Data Center Resources for Cloud Computing: A Vision, Architectural Elements and Open Challenges Cloud Computing and Distributed Systems (CLOUDS)*, Laboratory Department of Computer Science and Software Engineering, The University of Melbourne, Australia, 2010, 1–12.
[15] A Beloglazov, J Abawajy and R Buyya, Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centres for Cloud Computing, *Future Generation Computer Systems*, **2012**, 28, 755-768.
[16] R Buyya, A Beloglazov and J Abawajy, Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers, *Concurrency and Computation: Practice and Experience*, **2010**, 24(13), 1397–1420.
[17] RN Calheiros, R Ranjan, A Beloglazov, CAF De Rose and R Buyya, CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms, *Software - Practice and Experience*, **2010**, 41, 23–50.
[18] K Park and VS Pai, CoMon: A Mostly-Scalable Monitoring System for PlanetLab, *ACM SIGOPS Operating Systems Review*, **2006**, 40(1), 65–74.