Journal of Engg. Research **Vol. 5 No. (2) June 2017 pp.** *19-42*

كفاءة الطاقة لجهاز افتراضي بخوارزمية الهجرة

محمد الشايجي، سائد عبد و م. د. سامراجيش

قسم هندسة الكمبيوتر، كلية علوم و هندسة الحاسوب ، جامعة الكويت، ص.. ب. 5969 الصفاة 13060

الخلاصة

تعتبر استخدام موارد مراكز البيانات واستهلاك الطاقة من العوامل الهامة في مجال الحوسبة السحابية. إن الافتراضية تجعل الاستخدام الفعال لموارد أجهزة مراكز البيانات وذلك باستخدام الأجهزة الافتراضية .VM ومع ذلك، فإن معظم الطرق الحالية لا تنظر في خفض استهلاك الطاقة أثناء هجرة الأجهزة الافتراضيةVM. في هذه الورقة، نقدم خوارزمية هجرة جديدة حريصة على الطاقة للأجهزة الافتراضية (EVM) والتي تأخذ بعين الاعتبار عوامل حيوية مختلفة في خوادم مراكز البيانات أثناء هجرة الأجهزة الافتراضية. تستند تقنية ال EVM المقترحة على اختيار خادم (ESS) لتحديد الخادم الضحية ليتم اغلاقه. وتشير نتائج التقييم والمقارنات أن الخوارزمية المقترحة تخفض من الاستهلاك والضغط العام من حيث خفض عدد حالة التغييرات في الخوادم، وكذلك خفض هجرات VM والتذبذبات إضافة إلى تفوقها على الأساليب والطرق الموجودة مثل طريقة اختيار الخادم التعسفي (ASS) واستراتيجية الملائمة الأولى (FFS). تظهر النتائج التجريبية ان EVM حققت وفورات في الطاقة بنسبة 31 ٪ أكثر من ASS وبنسبة 15 ٪ أكثر منFFS وذلك عند تحميل الخادم 30 % من الحمل. إن تقنية EVM المقترحة يجب أن تعزز وتسهل استهلاك الطاقة في هجرة الأجهزة الافتراضية VM.

Energy efficient virtual machine migration algorithm

Mohammad H. Alshayeji*, Sa'ed Abed and Mault D. Samrajesh

Computer Engineering Department, College of Computing Sciences and Engineering, Kuwait University, Kuwait, P.O.Box 5969, Safat 13060

**Corresponding Author: Email: m.alshayeji@ku.edu.kw*

ABSTRACT

Datacenter resource utilization and energy consumption are important factors in cloud computing. Virtualization makes effective use of datacenter's resources by using Virtual Machines (VMs). However, most of the available methods for migrating VMs are oblivious to energy consumption. In this paper, we present Energy Efficient Virtual Machine Migration (EVM) technique in order to address pivotal factors that affect the datacenter's servers while migrating VMs. Our proposed EVM technique is based on Energy Based Server Selection (ESS) to select victim and target servers. Our comparative results show that EVM ameliorates the overhead in terms of lower number of server state changes, VM migration and oscillations, and outperforms existing methods, such as Arbitrary Server Selection (ASS) and First Fit Strategy (FFS). The experimental results show that EVM achieved 31% more energy saving than ASS and 15% more than FSS at 30% server load. The proposed EVM technique should enhance and facilitate energy consumption in VMs migration.

Keywords: Cloud computing; efficient energy conservation; load balancing; virtual machine; migration.

INTRODUCTION

Energy consumption of Datacenter is increasing rapidly, as more applications move towards cloud computing platforms. The energy cost of a typical datacenter doubles every five years (Buyya et al., 2010).Energy management for servers and datacenter operations has become a prime issue (Stoess et al., 2007).

Figure 1 shows a typical datacenter consisting of 'm' servers each with varied number of VMs. Most existing load balancing methods exclusively focuses either on server's performance aspect, or on the energy aspect. However, the challenge is how to strike a balance between performance and reducing energy consumption (Shen et al., 2016).

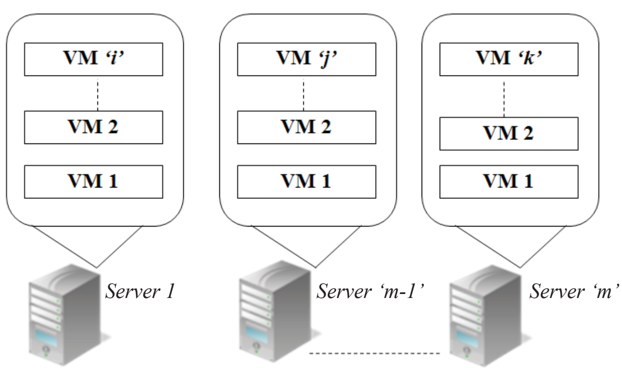


Fig. 1. A typical datacenter servers with virtual machine

Virtualization of server(s) at the datacenter consolidates resources and offers higher resource utilization, simplified resource management, and server maintenance. Moreover, virtualization enhances reliability and availability (Menascé, 2005). Virtualization software builds VMs that share the underlying hardware resources and function entirely as a discrete unit in the system. The ability to migrate VMs within datacenter servers can enhance resource utilization and minimize the energy consumption at datacenter (Al Shayeji & Samrajesh, 2012). Dynamic voltage/frequency scaling (DVFS) feature available in most modern CPU(s) conserves energy used by CPU based on its workload (Chen et al., 2008). However, a server in standby mode with no load requires about 66% of its maximum energy requirement (Fan et al., 2007; Ardagna, 2012). The energy is necessary to run important software services and power-up essential hardware components of the server.

Relocating VMs, based on the resource constraints is a bin-packing problem. A collection of items (VMs of diverse capacity) are required to be placed in bins (heterogeneous servers) such as to reduce the number of bins (servers) required. The ideal solution of a bin-packing problem is considered an NP-hard problem (Takeda & Takemura, 2010). A typical datacenter consists of hundreds of servers that are heterogeneous with varied computing capacities, energy requirement and number of VMs. It is challenging to find an allocation that is optimum. Therefore, there is a need to find methods that guarantee performance improvements by matching improvements in energy efficiency (Alahmadi et al., 2014; Patterson, 2008). The aim of this work is to develop a heuristic technique that satisfies multiple system constraints of server performance, meets the objective of reducing power consumption in datacenter by switching off underutilized servers, and strives to maintain the servers' load at the datacenter within the predefined thresholds.

Significant reduction in energy consumption is accomplished by relocating VMs from lightly loaded servers and turning them off. Migration or migration of VMs can be done either offline or live (Clark et al., 2005). Offline migration has some downtime, since it suspends and then resumes the operation. However, in live migration the downtime is hardly visible (Clark et al., 2005).

In this paper, we present a novel energy efficient virtual machine migration (EVM) technique that considers the server load, the number of VMs, the hit count, the energy requirement of the server and several other essential aspects in choosing a victim server. The victim server is filtered based on the aforementioned aspects and selected based on the highest energy first (HEF) strategy. The victim server VMs are relocated to target server(s) based on capacity and energy. The aim of the proposed algorithm is to reduce the energy requirement of datacenter and to balance the datacenter load among running servers. In addition, hit count is used in the selection of the victim server to avoid oscillation of VMs during migration. Figure 2 shows an overview of the proposed algorithm where EVM switches off under-utilized server C and its VMs are live migrated to either A or B based on the factors specified in the victim server selection algorithm.

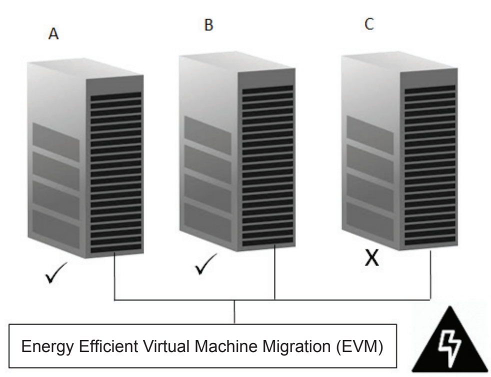


Fig. 2. General architecture of the proposed algorithm

The target server is selected considering the factors that reduce energy consumption using lowest energy first (LEF) strategy. At the victim server, when all the VMs are migrated, victim server is turned off to save energy. Additionally, when the overall datacenter load hits wake up threshold (WoT),where all servers are overloaded, additional servers are powered onto evade any performance drop. The load of each server is examined at regular intervals and underutilized servers, based on the EVM's defined criteria are moved to switch off state, thus minimizing the energy requirement of the datacenter.

A service level agreement (SLA) constraints based optimization of datacenter energy is presented in Goudarzi et al. (2012). The constraints are trade-off between the energy cost and client satisfaction in the system. The algorithm's goal is to reduce the energy cost, while satisfying the client-level SLAs in a probabilistic sense. However, the proposed solution had certain assumptions, such as predetermined VM performance levels that are not applicable in general SLA's contracts. Our proposed EVM algorithm is tested and evaluated by comparative study on various other strategies of VM migration such as arbitrary server selection (ASS) and first fit strategy (FFS).

The results show that the proposed algorithm outperforms existing methods, such as ASS and FSS, in energy conservation and reduces the carbon footprint while providing lower overhead (lower number of server state changes, VM migrations and VM migration oscillations).

The paper is structured as follows: Section 2describes the most related works. Our proposed EVM algorithms are presented in Section 3. The evaluation method of the proposed solution and discussion on the experimental results is presented in Section 4. Finally, conclusions and future works are in Section 5.

RELATED WORKS

Nowadays, computing done at datacenters has surged, resulting in increased energy consumption at the datacenter. Enhancing energy efficiency is a major challenge in cloud computing (Zou et al., 2014).Datacenter energy conservation is an eco-friendly approach that contributes to environmentally sustainable computing, and reduces energy cost. Several methods have been proposed in Wood et al. (2007), Singh et al. (2008) and Ye et al. (2010) to enhance the energy consumption of server elements including CPU, memory, networking components, and disk storage. However, with the extensive use of virtualization techniques in cloud infrastructures, customary energy-efficient approaches cannot be used at data center right away, due to the various levels of abstraction in virtualization (Al Shayeji & Samrajesh, 2012; Ye et al., 2010).

General VM migration methods

A preemptive migration for proactive fault tolerance is presented in Engelmann et al. (2009). This relocates parts of an application such as a process or VM from nodes that are about to fail. It uses feedback-loop control mechanism to monitor application's health based on pre-defined factors; increase in heat in the compute node is used to determine the health of the node. Computation or VM is relocated to other compute node. This reduces application breakdowns and extends application's mean time to failure (AMTTF). However, the datacenter's energy conservation is not considered.

Dynamic round-robin (DRR), an extended scheduling of the traditional round-robin scheduling, is presented in Lin et al. (2011). DRR incorporates the concept of state of retirement and retirement time threshold. In case a physical server is in its retirement state (a VM in the server completes its task, hence it is assumed that the left over VMs will also complete) and has VMs after the retirement threshold, the server will be forced to relocate its virtual machines and shutdown. It uses two rules to consolidate VMs; rule one avoids addition of extra virtual machines to a retiring server, so that it can be shutdown; rule two quickens the consolidation process for the shutdown of physical servers. However, the load of the server and VM during migration is not considered.

A VM migration framework that balances the load using the metric captured from physical server and the VMs is discussed in Arzuaga & Kaeli (2010). The algorithm used for balancing the load applies a greedy approach and predicts which VM migration will give the maximum enhancement of the imbalance metric. It can also predict the future server state. However, the server energy efficiency in the process of migration is not considered.

An online VM shuffling algorithm for VM migration is proposed in Engelmann et al. (2009). The paper formulated an optimization problem and showed that migrating VM is an NP-hard assignment problem. The proposed heuristic algorithm improves VM placement with controllable VM migration overhead. The VM migration algorithm attempts to minimize the time while shuffling. The algorithm is evaluated using real world and artificial traffic patterns. However, the energy conservation of datacenter servers is not considered in the algorithm.

Two strategies to balance the load with multiple virtual machines (VMs) using live migration are presented in Forsman et al. (2015). The push strategy migrate load to under loaded servers. The pull strategy is used by lightly loaded hosts to offload the overloaded nodes. The results showed that the pull strategy re-distributes the load, when the load is in the range low-to-medium. On the other hand, the push strategy is quicker for medium-to-high load. However, the re-balance time taken is higher.

A technical survey on live virtual machine migration technique is presented in Malleswari et al. (2015). The paper describes various virtual machine migrations and compares various methods based on several factors such as selection of target server, selection of source server, metrics considered, benefits and drawbacks. The paper concludes that gang migration reduces network downtime.

Energy based VM migration methods

A VM consolidation method for power saving in datacenter is presented in Takeda & Takemura (2010). The VMs are relocated to other server based on a unique rank assigned to each physical server, using extended first-fit decreasing (FFD) algorithm that considers one physical source server and one destination server for migration. Moreover, multiple concurrent VM migration is not permissible to a physical server and empty servers are kept powered on to avoid excessive power operations.

Heuristics reallocation of VMs to minimize energy using enhanced best fit decreasing order based on the current utilization of servers is presented in Beloglazov & Buyya (2010). The VMs are arranged in diminishing order based on their current utilization and VMs are allocated to host, which provides the smallest increase in energy consumption. It has three policies for selecting VMs for migration; 1) Migrate least number of VMs to reduce migration overhead, 2) Migrate VMs with least usage of CPU and 3) Migrate VMs randomly. However, individual VM migration to a server and its impact on future migrations are not considered.

An energy saving VM scheduling policy for clusters is proposed in Hu et al. (2008). VMs are combined on a multilayer ring-based overlay network. A detector is engaged to monitor the evil states of the clusters including node thrashing, job blocking, and fault tolerance state due to physical or software malfunctions; a controller is used to respond to the evil states. Migration action squeezed migrated VMs that are in the outer ring layer; migration action release would migrate VMs to an underutilized node. However, the network topology used is too complex.

An integrated VM migration algorithm that considers the server's load is presented in Al Shayeji & Samrajesh (2012). The algorithm is evaluated under VM's static load and dynamic load. All VMs from the selected victim server are migrated to a target server as a group. However, the datacenter consists of only homogenous servers. The energy requirement of each server is not considered during migration and VMs are not migrated individually to a suitable energy efficient server.

An energy-efficient storage migration of virtual machines is presented in Graubner et al. (2013). It uses a multi-layered root file system (MLRFS) and live VM migration technique. The implementation is done using Eucalyptus tool. The elementary images of VMs are distributed centrally and cached on a local disk. It has a separate layer for storing local changes that functions based on copy-on write mechanism. Only local alterations are transmitted during the process of disk synchronization. However, having VM images that have frequent updates affect server performance. Moreover, having multiple images reduces the chances for switching off servers and conserving energy.

An energy based virtual machine scheduling for cloud data centers is presented in Ghribi et al., 2013). The proposed solution includes two algorithms; the exact allocation and the migration algorithm. The paper claims that by combining the allocation algorithm with migration algorithm, energy can be saved. The proposed algorithm is compared with best fit algorithm. However, the presented solution considers only homogenous servers.

A green-power-aware virtual machine migration strategy is presented in Wang et al. (2015). The datacenters are powered by mixed supply of both grid and renewable energy. The paper concludes that the unique characteristics of datacenter workloads can be used to reduce its reliance on traditional energy sources. The proposed dynamic load balancing (DLB) aims to dynamically balance the workload on different physical machines. Moreover, joint optimal planning (JOP) optimizes the VM migration strategy towards exploiting the utilization of green energy.

The method in Wang et al. (2015) is closer to our work and aims to minimize the energy cost of the datacenter. However, the performance impact on the servers is not evaluated. Our proposed VM migration technique applies a hybrid approach by considering both energy requirement of servers and the load of the servers. The proposed algorithm effectively relocates VMs such as to minimize the overall datacenter energy requirement and provides effective utilization of datacenter resources. Moreover, our proposed algorithm migrate individual VM to suitable servers rather than migrating all VMs as a block from the victim server.

PROPOSED SOLUTION

The proposed energy aware VM migration algorithm strives to minimize the overall energy requirement of the datacenter, while maintaining the system performance. EVM technique incorporates three algorithms. The first algorithm victim server selection algorithm (VSS), selects the server to be switched-off. The second algorithm target server selection algorithm (TSS) selects the server to migrate the VMs. Finally, the datacenter performance is not degraded as the server load balance algorithm (SLB) balances the load and wakes up additional server when needed.

The following subsections describe the system model, objective function and the proposed dynamic EVM technique algorithms. Moreover, illustrative examples considering various scenarios of server states are also presented.

System model

The system model represents the number of physical servers in the datacenter (m)and the number of VM's currently residing on various servers in the datacenter (n).Each server has varied number of VMs, the VM count is done using VC_i. The VM's resource requirement is computed as in Chen et al. (2011) using virtual machine j resource requirement at $server_i$ ($VMRij$).VMR is computed using the CPU, memory, and network requirement of the residing application in the VM. The total load of the server is determined by the summation of VM's resource requirement represented as PLi. We consider heterogeneous servers, each server based on its capacity a hit threshold (HcT) is specified. Power off threshold (PoT) is used to switch off servers, wake up threshold (WoT) is used to wake up servers, when server load surges. Target server threshold (TsT) is used for selecting target server for placing the VMs. MPWi represents the energy requirement of ith server.

Below, we summarize the main terminologies used in the proposed technique.

m	: Number of physical servers
n	: Total number of VMs
q	: Number of physical servers running
R_i	: List of servers currently running $\{ 0\leq i\leq q \}$
SL_i	: List of servers switched off (sleeping) $\{ 0\leq i\leq m\text{-}q \}$
VC_i	: Virtual machine Count at server '*i*' $\{ i=1,2..q \}$
VMR_{ij}	: Virtual Machine '*j*' Resource requirement for server '*i*' as in (Chen *et al*., 2011; Wood *et al*., 2009)
PL_i	: Load at server '*i*'
PoT	: The Power off Threshold
WoT	: The Wake up Threshold
TsT	: The Target server Threshold
WsT	: The Datacenter wakeup server Threshold
HcT_i	: Hit count threshold for server '*i*'
HC_i	: The Hit Count of server '*i*'
T_i	: List of target servers for candidate list $\{ 0\leq i \leq q \}$
MPW_i	: Energy requirement of server '*i*'

Objective function

The energy function (Z) aims to minimize the energy consumption at the datacenter by switching off underutilized server(s) those satisfy specific constraints. This reduces the overall energy requirement of the server. The datacenter consists of '*m*' number of servers and each server holds VC_i set of VMs. The overall energy (Z) is defined as below:

$$Z = \sum_i MPW_i \,.\, S_i \quad 0 \le i \le m \qquad \ldots\ldots\ldots\ldots\ldots\ldots \tag{1}$$

$$S_i = \begin{cases} 1 \text{ if server } `i' \text{ is up and running} \\ 0 \text{ otherwise} \end{cases}$$

Where '*Z*' is the overall energy requirement of the datacenter observed at time 't'. The objective is to minimize '*Z*'.

Constraints

The objective function aims to minimize the energy consumption in the datacenter subject to the following constraints. The lower resource constraint ensures that the running server is never underutilized. Hence, only servers with sufficient load (> PoT) is powered on and running. The upper resource constraint ensures that the servers are not overloaded. The VM placement constraint guarantees that a VM is placed in only one server after on creation and after migration. Finally, the energy constraint defines that the optimization is performed only on active servers.

1. Lower resource constraint

$$\sum_i VMR_{ij} \,.X_{ij} >= PoT \,\forall j \tag{2}$$

$$X_{ij} = \begin{cases} 1 \text{ if VM } `j' \textit{ placed in Server } `i' \\ 0 \text{ otherwise} \end{cases}$$

The sum of VMs at server '*i*' at any time is always greater than the PoT. VMR derived considering the CPU, memory, and network requirement of the residing application in the VM.

2. Upper resource constraint

$$\sum_i VMR_{ij} \,.X_{ij} <= WoT \,\forall j \tag{3}$$

$$X_{ij} = \begin{cases} 1 \text{ if VM } `j' \textit{ placed in Server } `i' \\ 0 \text{ otherwise} \end{cases}$$

The load of any server is less than wake up threshold (WoT);otherwise, when a target server is available for migrating VMs based on TsT, selected VM's of the server are migrated to a target server. Moreover, in case all the servers are above WoT, new servers are powered off to balance the datacenter server's load.

3. Placement constraint

$$\sum_j X_{ij} = 1 \,\forall\, i \tag{4}$$

A VM is placed and resided in only one server at a time. When a VM is migrated and placed in a target server $Xij=1$, the victim server's value is set to 0, hence the summation is always 1.

4. Energy consumption constraint

$$MPW_i >= 0 \quad (5)$$
$$0 \leq i \leq q$$

The energy requirement of any server cannot be less than zero. The minimization of energy is performed only on servers up and running.

The lower resource constraint and upper resource constraint ensures that the server(s) strive to operate within the specified thresholds, while minimizing the energy consumption. Equality constraint on VM placement ensures that live migration of VM is performed and the VM resides at one server at a time. Finally, the energy consumption constrain ensures that minimization happened on energy is performed on running servers.

The proposed energy efficient virtual machine migration (EVM) technique

The proposed EVM technique consists of three algorithms; the victim server selection algorithm (VSS), target server selection algorithm (TSS) and the server load balance algorithm (SLB) described in the next sub-sections.

Victim server selection algorithm (VSS)

The victim server selection (VSS) algorithm is carried out by considering the current server load, the number of VMs, the threshold hit count, and the energy requirement of the server as shown in Algorithm1.

The algorithm is triggered, when one or more of the following conditions are satisfied: (a) A change in resources availability at the datacenter, (b) Departure of one or more VMs, (c) Dynamic VMs load increase or decrease. The VSS algorithm makes a list of physical servers that are lower than the *PoT*. Lists of servers maintained are of victim server list 'C', target server 'T'

A*lgorithm 1:LocateVictimServer() // Victim Server*

Output : Victim Server Selection and VM's to Migrates

Input : VM's Resource Requirement, PM's Load and Energy

Trigger : Change in Resources Availability, Arrival or Exit of VM Begin

1. While (Change in Datacenter Resources, VM arrival, Exit)

2. Find server(s) ∃PLi<PoT∧HCi<HcTi

3. Sort List 'C' descending based on Energy // HEF (Highest Energy First)

4. Choose the first Server from 'C'

// In case of similar choose based on VMR

5. Call LocateTargetServer()∀VCi individually End

Algorithm 1. Victim server selection (VSS)

The servers are monitored for a predefined time interval and each time when a server goes below *PoT*, the count HCi of the corresponding server hit count is incremented. The victim server is selected based on the server energy requirements using highest energy first (HEF) strategy such that maximum energy is conserved. Subsequently, the algorithm invokes LocateTargetServer()function and the VMs in the victim server are selected for migration as shown in Algorithm1. Each VM in the victim server is migrated separately to the target server(s) such that, each individual VM is placed in the best possible target server to minimize overall datacenter energy consumption. Moreover, the algorithm reevaluates the current server load at the datacenter before subsequent migration.

Target server selection algorithm (TSS)

The Target Server Selection (TSS) algorithm selects servers that are within the target server threshold (TsT) range and have the VM's resource requirement as shown in Algorithm 2. The algorithm populates to list all servers above TsT and such that VM's resource requirement (VMR) is satisfied. A least energy first (LEF) strategy is used to choose the target server from the list, such that it minimizes the energy requirement at the datacenter. Live migration technique is used to relocate the VMs to the selected target server. When all the VMs of the victim server are relocated, the victim server is powered off. TSS is called for every VM in the victim server, and respective lists are updated accordingly after each migration. In case of a change in the target server load that prevents VM migration, TSS attempts to find a target in the next cycle, while the VM continues to reside in the existing server and performs its operations.

Server load balance algorithm (SLB)

In case the overall datacenter's load goes beyond a pre-defined datacenter wakeup server threshold (WsT), SLB filters the servers, such that $PL_i > WoT$. Additional servers are woken up based on $VMR_i >$ Minimum capacity requirement (MCR) and LEF. Next, the VMs of the overloaded server is live migrated to the new server(s).

Algorithm 2: LocateTargetServer() // Target Server

Output : Target Server to Relocate VM's, Power off Victim Server

Input : VM's Resource Requirement

Trigger : Call from LocateVictimServer() Begin

1. For all Server up and running in the datacenter do

2. Find server $\exists$ *{ PLi>TsT }*$\wedge$*{Server Capacity} >* $\sum$*VMRi*

3. Sort List 'T' ascending based on Energy requirement

// LEF (Least Energy First)

4. Choose the first server // Least Energy First

5. VM is Live Migrated to the Target Server.

6. If (VCat victim Server = ϕ*)*

Switch off the selected victim Server End

Algorithm 2. Target server selection (TSS)

Algorithm 3 shows the server load balance algorithm (SLB), when an individual server's load goes beyond WoT; the servers are monitored for a period. If the load continues to hit WoT, then selective VM migration is performed to other running servers that have load above TsT and below WoT. In the case of the addition of new servers to the datacenter (assuming new servers are more Performance and energy efficient (PEE)), VMs from existing high energy servers (HES) are gradually migrated to the new server, thereby reducing the datacenter's overall energy requirement.

Server load thresholds

The proposed algorithm strives to keep the heterogeneous servers of the datacenter operate between the two server specific thresholds PoT and WoT (Wang et al., 2014; Sahu et al., 2013). PoT is used to list the candidate servers from which we select a victim server whose VMs are migrated and switched off. The target server is selected using target server threshold (TsT).

Migrations are done such that after migration the server does not gets overloaded, at the same time improve the chances of servers being switched off (i.e. don't choose servers that are underutilized).

Algorithm 3: Wakeup () // Wake up Server

Output : Waked up Server with Migrated VM's

Input : Minimum Capacity Requirement (MCR)

Trigger : Overall Datacenter Load above threshold Begin

1. *For all Server up and running do*
2. *If (OL>WoT) // Overall Data center Load (OL) is higher*
3. *For all Servers sleeping do*
4. *Find servers ∃VMRi> MCR*
5. *Sort SLi ascending on Energy requirement, // LEF (Least Energy First)*
6. *Choose the first server // Least Energy First*
7. *Live Migrate the VM from overloaded servers*
8. *else*
9. *Find individual server ∃PLi>WoT // Individual Server Load higher*
10. *If exists then Monitor PLi and Live Migrate VM*
11. *Call LocateTargetServer() such that PLi<WoT End*

Algorithm 3. Wake up server

WoT is the wakeup server threshold, when individual server WoT hits for defined HC, selective VM migration is performed. However, when the overall load of all servers in the datacenter is above WsT for specific time, additional servers are switched-on to balance the load in the datacenter.

Illustrative examples

To illustrate the process involved, we use simple scenarios to show how the migration and servers are powered off. Figure 3 (a) and (b) show that server S3 goes beyond WoT for a predefined time, selective VM migration is performed to existing powered on server S8. The scenario in Figure 3 (c) and (d) show that server S3 goes beyond WoT for predefined time. It has 2 VM that needs to be migrated. VM9 is migrated to S6. However, VM8 is migrated to S1 as the energy consumption for S1is less than S8.

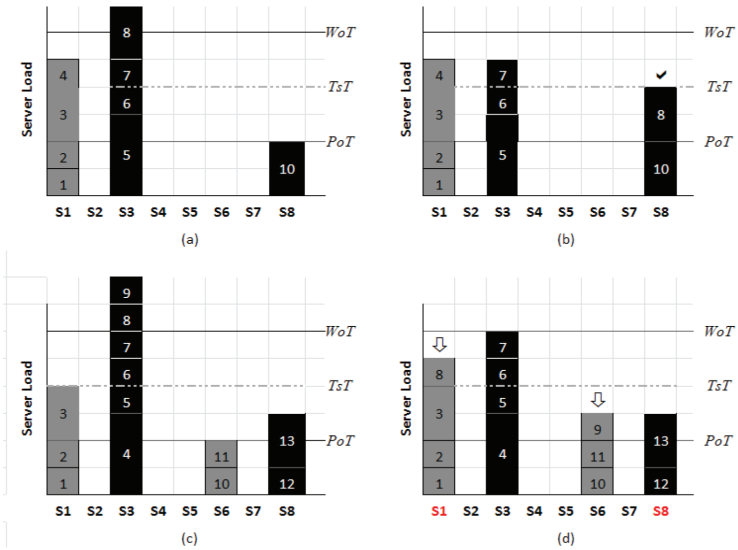


Fig. 3. Illustration of server states load balancing scenario-1. a)S3above WoT b) VM-8 Migrated from S3 to S8. Scenario-2 c) S3 above WoT d) VM’s of S3 VM-8 Migrated to S1, VM-9 Migrated to S6

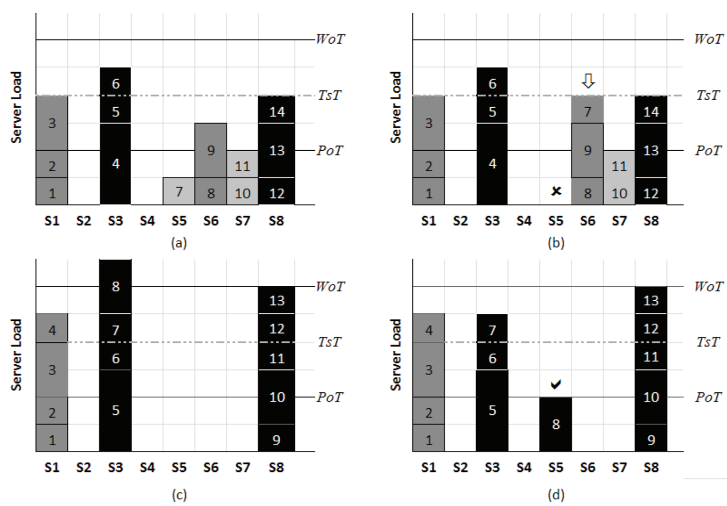


Fig. 4. Illustration of server states power on/off a) Server5 (S5) under PoT b) S5 is switched off and VM-7migrated to S6, S6 under TsT c) Overall Load (OL) of datacenter above WoT for predefined time d) VM-8 from S3 migrated to S5 newly switched on

Figure 4 (a) and (b) shows an illustrative example, where server S5 goes below PoT and S5 switched off. WoT (wake up threshold) is used to migrate VMs in Figure 4 (c) (d) when the overall datacenter load goes beyond WoT the selected VMs of a server S3 are migrated to the newly switched on server S5 such as to balance the overall datacenter servers load.

PERFORMANCE EVALUATION AND DISCUSSIONS

We evaluate our proposed technique by modeling the datacenter servers of both heterogeneous capacity and heterogeneous energy consumption. Changes in server load are based on the servers' VM resource requirements. The VM's load is based CPU, memory, and network requirement of the residing application. The architecture and the initial setup of the simulator are described in the following subsections. Moreover, we focus on switching off servers, as standby servers with no load require about 66% of its maximum energy requirement.

Architecture of the Simulator

Simulation model

The architectural design of the simulator consists of two main modules, explicitly the Server Load Generator (SLG) and EVM (Al Shayeji & Samrajesh, 2012). SL generates server load and initializes the VMs. VM's load is based CPU, memory, and network requirement, modeled by normal distribution using Box-Muller transformation as in Akoush et al. (2010; Box & Muller (1958) and Koomey (2007). The load of the server generated signifies a typical datacenter load. EVM observes the server's load; the number of VMs and hit count. EVM then migrates VMs from the selected victim server such that server can be switched off to decrease energy consumption in the datacenter. Figure 5 shows the control flow of the model. Changes in server load is based on the VM's resource requirement. The various events that trigger change in server load are VM's arrival, VM's exit, VM's load expand (i.e. VM's load increases, typical scenario include spike in users connect to an application running as VM),VM's load shrinkage (i.e. VM's load reduces, typical scenario include slide in users connected to an application running as VM). Moreover, change in datacenter server(s) triggers VM migrations. The relocation of VM(s) is done such as to minimize energy consumption of the datacenter.

The simulation setup parameters values are given in Table 1. Datacenter consists of 100 servers that have heterogeneous capacities and consume diverse energy. The servers are always connected with highly available network. Moreover, we consider that 1000 VMs and each VM have heterogeneous resource requirements. The maximum number of VMs per server is ten. The various thresholds such as power off threshold (PoT), wake up threshold (WoT), target server threshold (TsT) is 30%, 85% and in the range 45% - 70%, respectively. Live migration is applied for VM migration across servers.

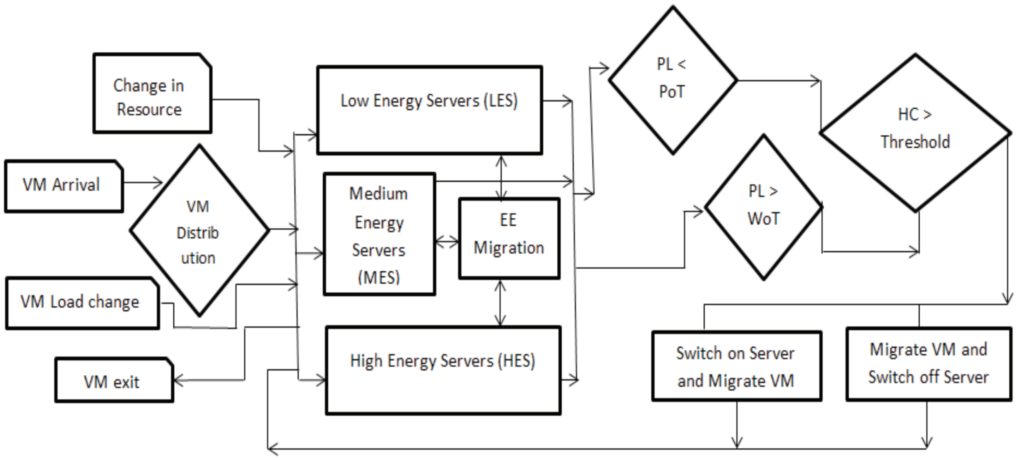


Fig. 5. Simulation model of a heterogeneous datacenter

We use the energy equation that connects power and time as given in Equation (6).

$$\text{Energy} = \text{power x time} \quad \ldots\ldots\ldots\ldots.. \qquad (6)$$

The energy consumption and energy conservation of the datacenter servers is expressed and measured in terms of kilo Watt in an hour(kWh), where kW is the power and h is the time in hours.

Table 1. Setup Paramters

Parameter	Value
Number of heterogeneous physical servers (N)	100
Total no of VMs 1000 (n)	1000
Max number of VMs per server (m)	10
Power off Threshold (PoT)	30%
Wake up threshold (WoT)	85%
Target server threshold (TsT)	45% - 70%

The energy requirements of servers are classified into low energy servers (LES), medium energy servers (MES), and high energy servers (HES). The energy requirement is given in Table 2 based on the estimated power consumption of servers in the U.S. and rest of the world (Koomey, 2007). The average energy requirement of the three types of server is as follows; for LES it is 218,638 for MES and 12682 kWh for HES.

Virtual machine management

VM arrival and exit at the datacenter is modeled based on Poisson distribution (Kim et al., 2009). The server load varies based on the current number of VMs the server holds and the load of each VM. Each VM's load is dynamic such as to represent a typical datacenter. The load at each server, number of VMs it has is monitored and the proposed algorithm is called to examine the opportunity of relocating VMs such as to switch-off server. When a new VM arrives at the datacenter for server allotment, assignment is done considering the performance-energy trade-off of the server. Datacenter servers that are above the target server threshold (TsT) and below the WoT (wake up threshold) are contenders for the newly arrived VM.

Table 2. Datacenter heterogeneous server energy

Server Classification	**Server Brand and Model**			**Average Total Energy Requirement (kWh)**
Low energy servers (LES)	HP DL380	Dell 2850	HP DL360	218
Medium energy servers (MES)	IBM i5-520	IBM p5 570	Sun V490	638
High energy servers (HES)	IBM p5 595	HP SDOME	Sun E25K	12682

Victim and target server selection

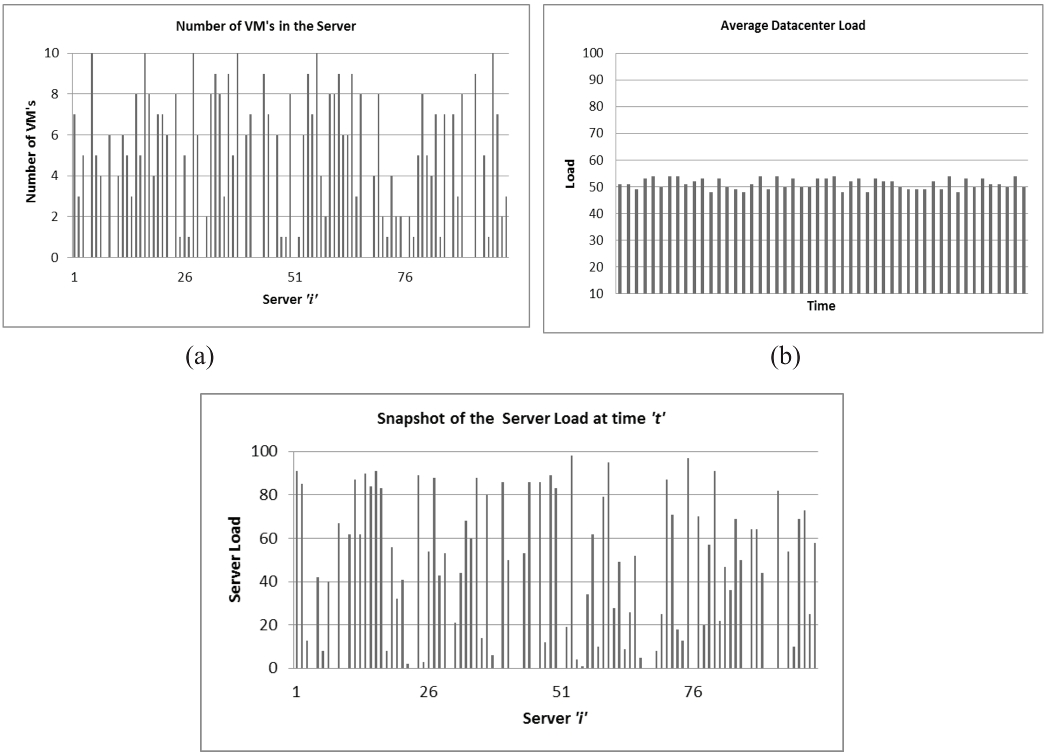
The selection of victim server and target server plays an important role in the process VM migration. The main objective of the VM migration is to reduce energy; incorporating efficient selection strategies further minimizes the energy requirement of the datacenter. In the proposed EVM technique, the victim server is selected based on HEF, candidate target servers are determined based on TsT and using LEF strategy the target server for migration is selected for each individual VM in the victim server. When all the VMs in the victim server are relocated, the victim server is switched off. Thereby reducing the datacenter overall energy consumption.

Initial evaluation setup

The setup parameters for the simulation are given in Table 1. The initial setup for datacenter includes 100 heterogeneous servers and 1000 VMs. The thresholds are based on individual server capacity; TsT is between the minimum and maximum values mentioned. Hit count (HC) is initially set to 10, it can be modified for each server. The datacenter consists of heterogeneous servers of three types: namely low range, medium range and high-end servers based on their energy requirement.

The energy requirement of each server varies based on Koomey (2007) as shown in Table 2shows the heterogeneous datacenter simulation model generated by Kelton et al. (2002).

The average number of VMs allotted to servers is shown in Figure 6 (a), the snapshot of the server load at time 't' is shown in Figure 6 (b). Newly switched-on server's load varies based on the migrated VM's resource requirement. Average datacenter load observed is shown in Figure 6 (c) which clearly indicates that the datacenter servers are underutilized.



(a) (b)

(c)

Fig. 6. a) Average number of VM's Alloted to Servers b) average datacenter load observedc) Snapshot of the server load at time 't'

Discussions

We perform comparative evaluation on the proposed EVM technique against arbitrary selection strategy (ASS) (Beloglazov & Buyya, 2010) and first fit strategy (FFS) (Goudarzi, et al., 2012). Both algorithm migrate VMs such as to reduce energy consumption at datacenter, which makes them best suitable for our comparative evaluation. We apply the above strategies in choosing the victim and target servers. We analyze the overall number of migrations, number of servers switched off, server state changes, energy consumption and the reduction of carbon footprints. The servers in the datacenter are arranged based on their server ID. In case of FFS, the victim and target servers are selected from the available list that first fits the VM's resource requirement. In case of ASS, the selection is based on random choice in which a random server ID is generated based on the available list of servers, if the VM requirement fits in the server it is selected, else the server ID is regenerated and the process repeats.

Network complexity and computational complexity

We use the Network Complexity Index (NCI) based on Drzymalski (2015) to compute the network complexity using the size and number of the sub-networks. Given that there are 'c' sub-networks in a network *N*, and *X[1]*, . . . , *X[c]*, *X[i]* denotes the size of the each sub-network. The NCI of the network N is defined as *NCI(N) = Max (X[i])*.

The proposed EVM algorithm works on servers connected in a datacenter with heterogeneous server capacities. The network complexity is limited to the datacenter sub-networks. Hence, the network complexity NCI(N) is minimum.

We estimate the computational complexity function using Big O notation. The computational complexity of the algorithms are as follows for EVM and FFS has O(n) while for ASS it is $O(n^2)$. EVM uses radix sort hence the computational complexity is similar to FFS.

Migration count

The number of VM migration at datacenter for the load of 30%, 50%, and 70% are analyzed under ASS, FFS and the proposed EVM algorithm, respectively. Figure 7 shows the number of VM migrations performed using the above strategies. Initially, the migrations are higher as more VMs are migrated to switch-off servers due to a higher number of servers hitting the PoT. The number of migrations tends to be minimal, when the load is at 50% since fewer servers hit the thresholds and the migration process is triggered less frequently. The number of migration tend to increase again as the datacenter server load approaches 70% as more servers hit the higher threshold (WoT) and this triggers the migration process. The number of migrations in EVM is lower since it uses various factors in the decision to migrate. A less number of VM migration results in reduced migration overhead.

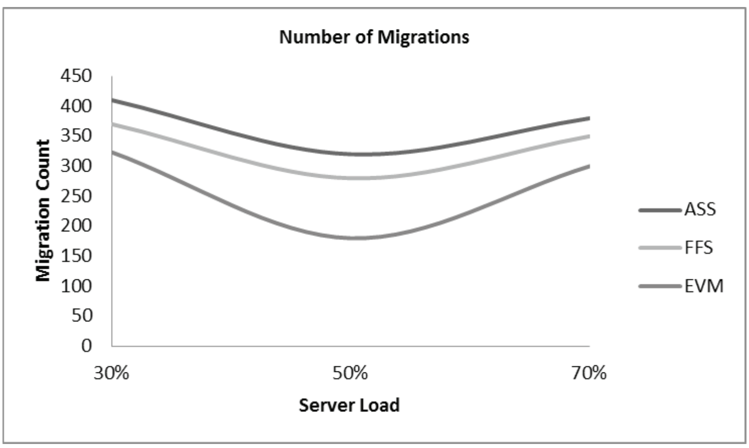


Fig.7. Number of VM migrations

The number of migration is consistently less for all server loads for the proposed EVM, at 30% server load. EVM offers 27% less migrations than ASS and 15% less than FFS. Moreover, at 50% datacenter server load, EVM is 78% better than ASS and 56% better than FFS.

Number of servers powered off

The number of servers switched off at 30%,50%,70% is shown in Figure 8. ASS, FFS, and EVM attempts to keep servers between the thresholds (30%-85%). At 30% server load, EVM switches off 31% more servers than ASS and 15% more than FFS. At 70% server load, EVM switches off 67%, 42% more servers than ASS, FFS respectively. The comparative percentage of servers powered off is higher at 70%server load as the number of servers switched off tends to decrease.

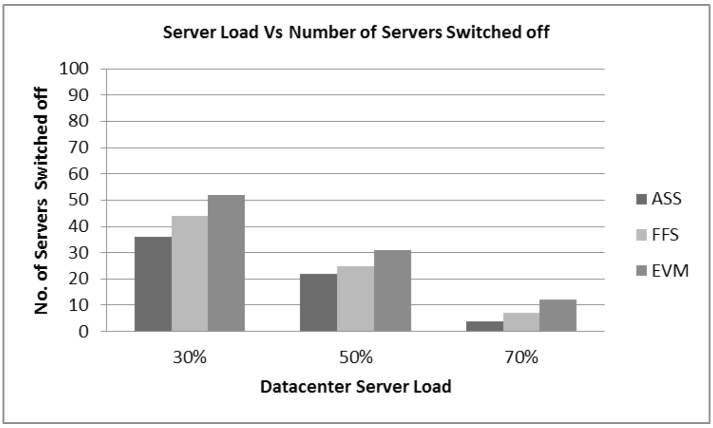


Fig. 8. Average server load and number of server(s) switched off

Maximum numbers of servers are switched off at 30% server load when the server is underutilized and algorithms migrates VMs and switches off servers. The proposed EVM out-performs other strategies under all the above server loads. When the overall datacenter load is above 85% and towards 100%, all available servers are running and server switching-off opportunities are minimal.

Server state changes

The load at each server varies, based on the VM's resource requirement. Server state changes between on and off based on VM migrations and the decision to switch off servers when load is less than PoT and the decision to switch on when overall load goes beyound WoT. The higher the number of state changes leads to an increased overhead in starting and shuting down servers. Figure 9 shows the number of server state changes (switch on and off). The algorithm's main aim is to conserve energy. In the case of addition of new servers to the datacenter, assuming new servers are more performance and energy efficient (PEE). VMs from exsiting high energy servers (HES) are gradually migrated to the new server therby reducing the datacenter's overall energy requirement.

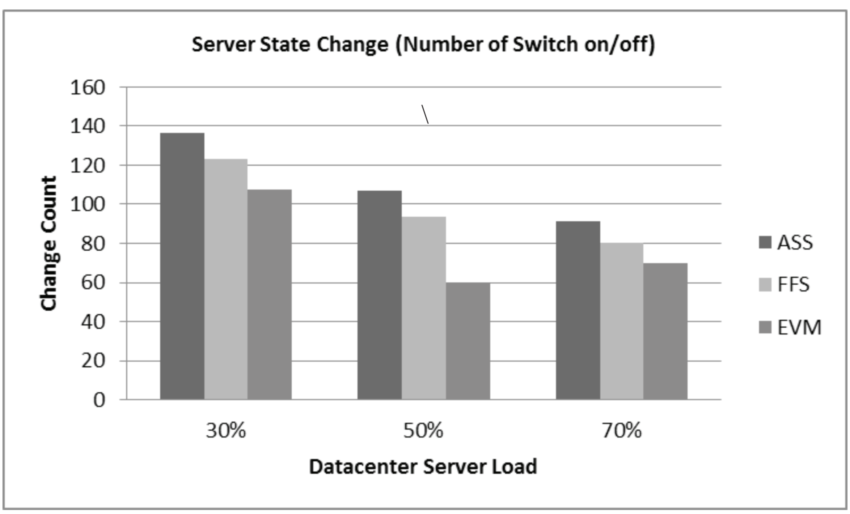


Fig. 9. Number of server state changes

The chart shows EVM has lower number of server state changes. Moreover, from Figure 7, we find that the number of servers switched-off is higher in EVM. This implies that the servers remain in switched-off state for a longer duration. EVM considers various factors during the server state change process (i.e. switch on-off) thus minmizing the number of future server state changes. At 30% server load, EVM offers 27% less server state changes when compared to ASS and 15% less server state changes, when compared to FFS. The comparative percentageof number servers state changes is higher at 50% load. EVM has 78% and 56% less server state changes when compared to ASS and FFS, respectively.

Datacenter load and energy conservation

A typical datacenter consists of varying energy consumingservers that are hetrogenous in nature. The proposed EVM uses energy based server selection (ESS) that applies highest energy first (HEF) strategy in choosing the victim server and lowest energy first (LEF) strategy for target server selection. This ensures that, while migrating VM's energy consumption of the victim, target servers are considered. At 30% server load, the energy savings achieved using EVM is 31% more than ASS and 15%more than FFS.At 70% server load, 67% more than ASS and 43% more than FFS. The proposed EVM algorithm conserves maximum energy when compared with ASS; FFS strategies for datacenter server load at 30%,50%,70% as shown in Figure 10. The increased saving of energy is possible because EVM is able to switch off more servers as shown in Figure 9.

Moreover, when EVM needs to switch on/off servers, it always chooses a server from the available list such as to maximize energy conservation. Table 3 shows the summary of various triggers, their corresponding action performed and the energy, performance implications. As shown in the table, EVM strives to maintain servers between PoT and WoT and conserves energy without any impact on the servers perfromance.

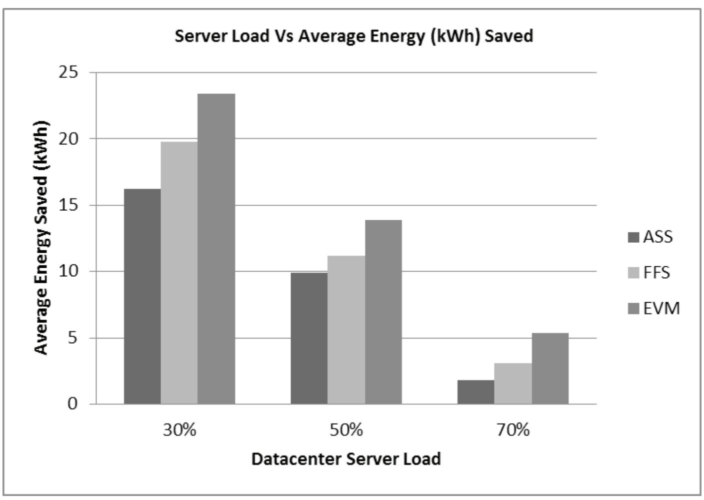


Fig. 10. Server load and average energy saved

This is possible since EVM observes the system for a predefined duration to ensure that a migration is warranted before VM migration is initiated, thus avoiding repeated VM migration or oscillation and thereby, reducing the migration count.

Table 3. Trigger and energy, performance implication

Trigger	**Actions Performed**	**Energy Consumption**	**Overall Performance**
Server goes below *PoT*	Migrate VMs from victim server and switch off victim	Decreased	No Change
Overall datacenter server load goes above WsT	Switch on Additional Server	Increased	Improved
Individual server WoT high for observed time	Migrate VMs	No change	Improved

Reduction in carbon footprint

Every kilowatt-hour (kWh) of energy conserved saves about a kilogram of carbon dioxide that is released to the environment and pollutes it. Carbon dioxide (CO_2) is the main contributor of global warming. Carbon footprint is a measure of the total amount of carbon dioxide emitted directly and indirectly due to an activity (Wiedmann & Minx, 2008). An activity here is a server at datacenter switched-on.Figure11 shows the reduction of carbon footprint under various datacenter loads based on the number of servers powered off. Emission factor is computed as 7.055×10^{-4} metric tons CO_2 / kWh. EVM offers significant reduction in carbon footprint at 30% server load. It offers 20% more reduction than ASS and 10% more than FFS. At 50% server load, EVM offers 13% more reduction than ASS and 5% more than FFS. The maximum reduction of carbon footprint is at 30% load as more servers are switched-off.

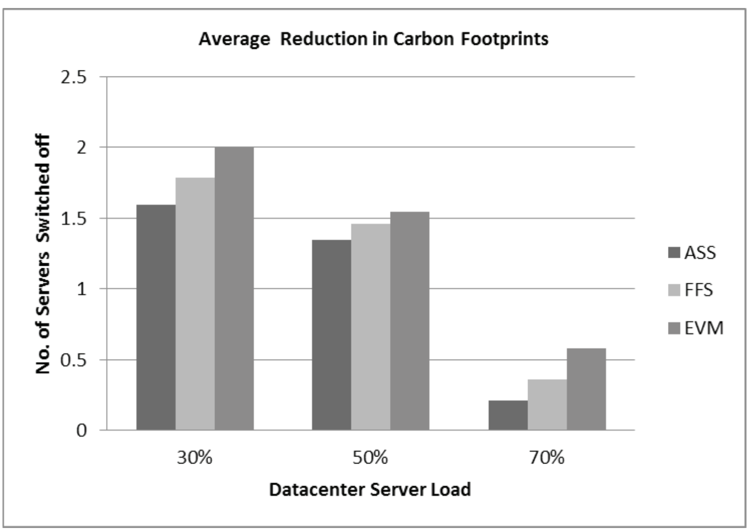


Fig. 11. Carbon (CO_2) footprint reduction

Case study on enterprise computing infrastructures

We performed an evaluation case study based on the energy consumption of an enterprise-style computing infrastructures using dataset in Stanford (2015). The dataset consists of energy consumption of network switches, thin clients, desktop computers, servers, and printers.

We applied our proposed VM migration technique on the dataset and studied the energy conserved. Figure 12 shows the average energy saved on the days of week.

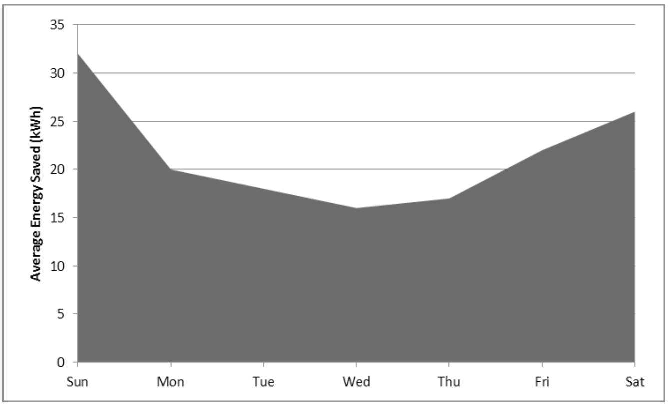


Fig. 12. Energy conserved in enterprise computing infrastructures

Further the observation shows that the enterprise server load gradually tend to increase until the middle of the week. The number of servers switched off in beginning and the end of week are more hence the energy conserved is also higher when compared with non-power aware (NPA) methods.

Statistical analysis

We apply the standard approach to test the results using null hypothesis (H0) against alternative hypothesis (HA) on the number of Server(s) switched off. We compute f(x) based on the Equation in (7) using our simulation dataset. The statistical values for the number of Server(s) switched off are given in Table 4. The mean is 30.3, the standard deviation (s) is 1.79 and the confidence interval alpha (α =0.05) is at 5%.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{7}$$

Table 4. Statistical values

Mean (m)	30.3
Standard deviation (s)	1.79
Alpha (α =0.05)	5%.

Hypothesize: The mean number of Server(s) switched off at 50% server load is greater than or equal to 28.

H_0: $\mu >= 28$ null hypothesis

H_A: $\mu < 28$ alternative hypothesis.

Figure 13 shows a region for the values analyzed for the hypothesis, the number of servers switched off and their Probability Density (PD). Based on the results, we concluded that the average number of servers powered off at 50 % servers load is less with significance level alpha (α =0.05) 5%. Hence, we accept the null hypothesis.

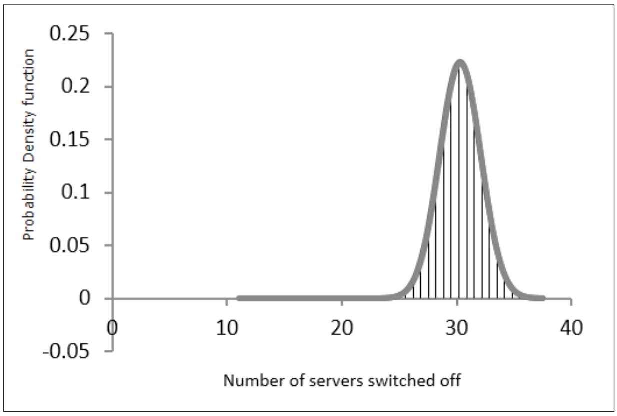


Fig. 13. Statistical analysis probability density (PD) function of number of servers switched off

CONCLUSION

We have presented a novel energy efficient virtual machine (EVM) migration technique for heterogeneous datacenter servers. EVM considers the datacenter server load and energy requirements of each server and applies energy based server selection (ESS) methods to select victim and target server using highest energy first (HEF) strategy and lowest energy first (LEF) strategy respectively. EVM strives to minimize the datacenter energy consumption by migration of VMs and switching off under loaded servers. We have analyzed the proposed solution using various factors such as the number of VM migrations, count of servers switched off, the number of server state changes, overall energy consumption and the reduction in carbon footprints. Our comparative evaluation and discussions shows that the proposed EVM technique is effective in minimizing the number of powered-on servers at a typical datacenter and reduces the overall energy consumption and carbon footprints. At 30% server load, the energy savings achieved using EVM is 31% more than ASS and 15% more than FFS. Moreover, EVM offers significant reduction in carbon footprint at 30% server load; the decrease is 20% higher than ASS and 10% higher than FFS. Our case study based evaluation on enterprise servers show that significant energy reduction is observed when compared with NPA. Additionally, our statistically analysis shows that the average number of powered off servers tend to behave as experimented.

Our future works include studying the server affinity and VM affinity and its impact in reduction of energy requirement at the datacenter.

REFERENCES

Akoush, S., Sohan, R., Rice, A., Moore, A.W. & Hopper, A. 2010. Predicting the performance of virtual machine migration. IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS):pp.37-46.

Al Shayeji, M.H. & Samrajesh, M.D. 2012. An energy-aware virtual machine migration algorithm. IEEE International Conference on Advances in Computing and Communications (ICACC): pp.242-246.

Alahmadi, A., Alnowiser, A., Zhu, M.M., Che, D. &Ghodous, P. 2014. Enhanced first-fit decreasing algorithm for energy-aware job scheduling in cloud. IEEE International Conference on Computational Science and Computational Intelligence, pp.69-74.

Ardagna, D., Panicucci, B., Trubian, M. & Zhang, L. 2012. Energy-aware autonomic resource allocation in multitier virtualized environments, IEEE Transactions on Services Computing, **5**(1): pp.2-19.

Arzuaga, E. & Kaeli, D.R. 2010. Quantifying load imbalance on virtualized enterprise servers.Proceedings of the ACM 1st Joint WOSP/SIPEW International conference on Performance engineering, pp.235-242.

Beloglazov, A. & Buyya, R. 2010. Energy efficient allocation of virtual machines in cloud data centers. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid): pp.577-578.

Box, G.E. & Muller, M.E. 1958. A note on the generation of random normal deviates. The Annals of Mathematical Statistics, **29**(2) pp.610-611.

Buyya, R., Beloglazov, A. & Abawajy, J.2010. Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. arXiv:1006.0308.

Chen, G., He, W., Liu, J., Nath, S., Rigas, L., Xiao, L. & Zhao, F. 2008. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation: pp.337-350.

Chen, M., Zhang, H., Su, Y.Y., Wang, X., Jiang, G. & Yoshihira, K. 2011. Effective VM sizing in virtualized data centers. IEEE International Symposium on Integrated Network Management, pp.594-60.

Clark, C., Fraser, K., Hand, S., Hansen, J.G., Jul, E., Limpach, C., Pratt, I. & Warfield, A. 2005. Live migration of virtual machines. In Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation, pp273-286.

Drzymalski Julie, 2015. A measure of supply chain complexity incorporating virtual arcs. Journal of Systems Science and Systems Engineering, **24**(4):pp.486-499.

Engelmann, C., Vallee, G.R., Naughton, T. & Scott, S.L. 2009. Proactive fault tolerance using preemptive migration. IEEE 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing, pp.252-257.

Fan, X., Weber, W.D. & Barroso, L.A. 2007. Power provisioning for a warehouse-sized computer. In ACM SIGARCH Computer Architecture News, **35**(2): pp.13-23.

Forsman, M., Glad, A., Lundberg, L. & Ilie, D. 2015. Algorithms for automated live migration of virtual machines. Journal of Systems and Software, (101):pp.110-126.

Ghribi, C., Hadji, M. & Zeghlache, D. 2013. Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms. 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid): pp.671-678.

Goudarzi, H., Ghasemazar, M. & Pedram, M. 2012. SLA-based optimization of power and migration cost in cloud computing. 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid): pp.172-179.

Graubner, P., Schmidt, M. & Freisleben, B. 2013. Energy-efficient virtual machine consolidation. IT Professional, (2):pp.28-34.

Hu, L., Jin, H., Liao, X., Xiong, X. & Liu, H. 2008. Magnet: A novel scheduling policy for power reduction in cluster with virtual machines. IEEE International Conference on Cluster Computing, 13-22.

Kelton, W.D., Sadowski, R.P. & Sadowski, D.A. 2002. Simulation with ARENA. McGraw-Hill, Inc.

Kim, K.H., Beloglazov, A. & Buyya, R. 2009. Power-aware provisioning of cloud resources for real-time services. In Proceedings of the ACM 7th International Workshop on Middleware for Grids, Clouds and e-Science, pp.1-6.

Koomey, J.G. 2007. Estimating total power consumption by servers in the US and the world, Lawrence Berkeley National Laboratory, Berkeley. pp.1-5.

Lin, C.C., Liu, P. & Wu, J.J. 2011. Energy-aware virtual machine dynamic provision and scheduling for cloud computing. IEEE International Conference on Cloud Computing (CLOUD): pp.736-737.

Malleswari, T.N., Vadivu, G. & Malathi, D. 2015. Live virtual machine migration techniques-a technical survey. Advances in Intelligent Systems and Computing, (308): pp.303-319.

Menascé, D.A. 2005. Virtualization: Concepts, applications, and performance modeling. International Conference on Computer Measurement Group, pp.407-414.

Patterson, D.A., 2008. The data center is the computer. Communications of the ACM, **51**(1): pp.105-105.

Sahu, Y., Pateriya, R.K. & Gupta, R.K. 2013. Cloud server optimization with load balancing and green computing techniques using dynamic compare and balance algorithm. 5th International Conference on Computational Intelligence and Communication Networks (CICN): pp.527-531.

Shen, Y., Bao, Z., Qin, X. & Shen, J. 2016. Adaptive task scheduling strategy in cloud: when energy consumption meets performance guarantee. World Wide Web-Internet and Web Information Systems, Springer, pp.1-19.

Singh, A., Korupolu, M. & Mohapatra, D. 2008. Server-storage virtualization: integration and load balancing in data centers. In Proceedings of the ACM/IEEE conference on Supercomputing, pp.1-12.

Stanford PowerNet project, dataset available http://sing.stanford.edu/maria/powernet/ available online accessed on 10 October 2015.

Stoess, J., Lang, C. & Bellosa, F. 2007. Energy management for hypervisor-based virtual machines. In USENIX annual technical Conference, pp.1-14.

Takeda, S. & Takemura, T. 2010. A rank-based VM consolidation method for power saving in datacenters. Information and Media Technologies**, 5**(3): pp.994-1002.

Wang, J., Huang, C., Liu, Q., He, K., Wang, J., Li, P. & Jia, X. 2014. An Optimization VM Deployment for Maximizing Energy Utility in Cloud Environment. Springer International Conference on Algorithms and Architectures for Parallel Processing, pp.400-414.

Wang, X., Du, Z., Chen, Y. & Yang, M. 2015. A green-aware virtual machine migration strategy for sustainable datacenter powered by renewable energy. Simulation Modelling Practice and Theory, (58): pp.3-14

Wiedmann, T. & Minx, J. 2008. A definition of 'carbon footprint'. Ecological economics research trends, (1): pp.1-11.

Wood, T., Shenoy, P., Venkataramani, A. & Yousif, M. 2009. Sandpiper: Black-box and gray-box resource management for virtual machines. Computer Networks, **53**(17): pp.2923-2938.

Wood, T., Shenoy, P.J., Venkataramani, A. & Yousif, M.S., 2007. Black-box and Gray-box Strategies for Virtual Machine Migration. In USENIX Symposium on Networked Systems Design and Implementation, pp.17-17.

Ye, K., Huang, D., Jiang, X., Chen, H. & Wu, S. 2010. Virtual machine based energy-efficient data center architecture for cloud computing: a performance perspective. In Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, pp.171-178.

Zou, S., Wen, X., Chen, K., Huang, S., Chen, Y., Liu, Y., Xia, Y. & Hu, C. 2014. Virtualknotter: Online virtual machine shuffling for congestion resolving in virtualized datacenter. Computer networks, (67): pp.141-153.

Submitted: 2/April/2016
Revised : 27/Sept/2016
Accepted : 27/Oct/2016