Chapter 3
Multi-objective Dynamic Virtual Machine Consolidation Algorithm for Cloud Data Centers with Highly Energy Proportional Servers and Heterogeneous Workload

Md Anit Khan, Andrew P. Paplinski, Abdul Malik Khan, Manzur Murshed, and Rajkumar Buyya

Contents

M. A. Khan · A. P. Paplinski · A. M. Khan
Faculty of Information Technology, Monash University, Clayton, VIC, Australia
e-mail: andrew.paplinski@monash.edu; malik.khan@monash.edu

M. Murshed
School of Information Technology, Federation University Australia, Churchill, VIC, Australia
e-mail: manzur.murshed@federation.edu.au

R. Buyya (✉)
CLOUDS Lab, School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC, Australia
e-mail: rbuyya@unimelb.edu.au

R. Buyya et al. (eds.), *New Frontiers in Cloud Computing and Internet of Things*, Internet of Things, https://doi.org/10.1007/978-3-031-05528-7_3

3.1 Introduction

Data Center (DC) construction has increased 47% in 2017, resulting in consumption of 3% of the world's energy [1] – equivalent to the energy consumption by the airline industry [2]. Research on DCs of the United States has revealed that in 2014, the sum of energy usage by all DCs in the United States was 70 billion kWh – accounting for 1.8% of the country's total energy usage [3]. More importantly, the trend of energy consumption of DCs highlights that energy consumption is rising every year and is expected to increase by 4% from 2014 to 2020. The energy consumption by the DCs of the US data centers is estimated to reach up to 73 billion kWh in 2020. Subsequently, countries across the world have come forward to address the challenge of increasing energy consumption by DC [4]. Joint Research Centre (JRC) of European Commission has formed the Code of Conduct for energy efficiency in CDC with an aim to inform and encourage DC owners and operators to effectively level off the energy consumption [5]. Such recent literature highlights great significance of energy consumption minimization of CDC and its relevance to present day.

In 2018, JRC has proposed a detailed guideline in relation to the best practices to limit the energy consumption of DCs [6]. One of the practices strongly advised by JRC is called VM Consolidation (VMC). The essence of VMC is to consolidate VMs in minimum possible number of PMs, so that the number of unused PMs having no VMs can be maximized and turn them into lower energy consumption state, such as sleep state or turned off state. Therefore, energy consumption can be reduced regardless of the types of PMs and regardless of the types of energy sources used to power those PMs. The data representing VMRT (i.e., the lifespan of a VM) of real VMs resided in Nectar Cloud [7] highlighted through Figs. 3.1, 3.2, and 3.3 exhibit that VMRT varies from one VM to another, whereas existing DVMC algorithms, such as [8–23], either mentioned that homogeneous VMRT has been used for experiments or have not articulated any assumption made on VMRT. Through experimenting with homogenous VMRT (i.e., all VMs are assigned with tasks of equal length in terms of task finishing time) using CloudSim [24, 25] as used by respective researchers, we have obtained similar results as presented in respective literature. However, in real scenario, CDC consists of VMs with heterogeneous VMRT. In other words, VMs are assigned with tasks of unequal lengths in terms of task finishing time.

It is critical to note that VMRT has strong impact on CDC energy consumption. Larger VMRT increases CDC energy consumption, since the total resource utilization by a VM grows as the lifespan of that VM grows. Hence, without considering heterogeneous VMRT, the overall estimation about the impact of any DVMC algorithm on CDC energy consumption and subsequent QoS would be less accurate. In contrast with traditional DVMC algorithms which only consider homogeneous VMRT, *RTDVMC* takes the heterogeneous VMRT into consideration [26]. Nevertheless, several limitations exist with *RTDVMC* as elucidated in the following:

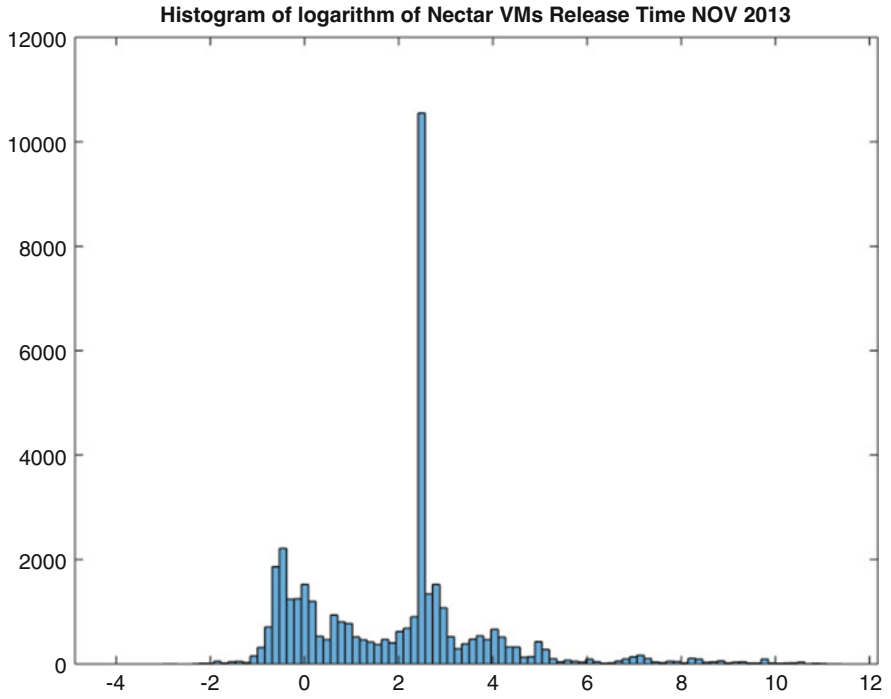

Fig. 3.1 Histogram of logarithm of release time (second) of VMs created in Nectar Cloud in November 2013

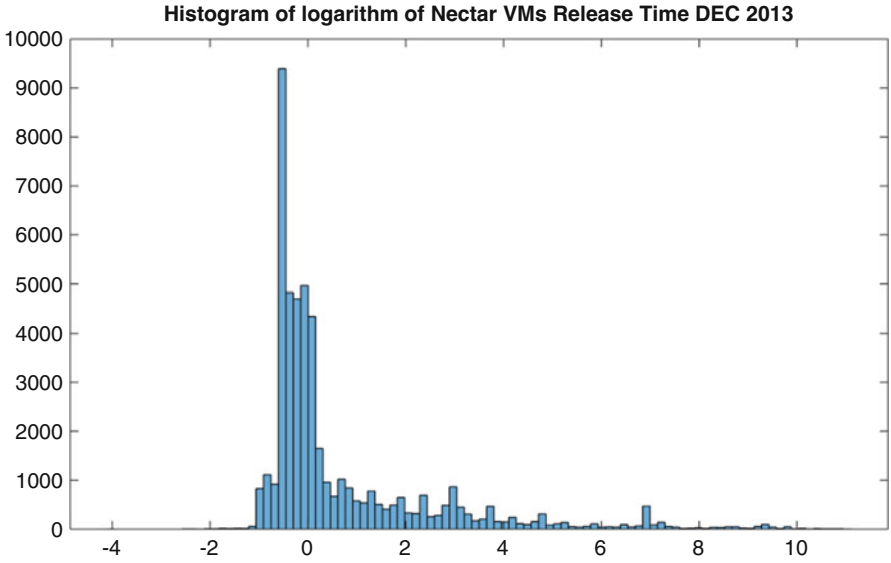

Fig. 3.2 Histogram of logarithm of release time (second) of VMs created in Nectar Cloud in December 2013

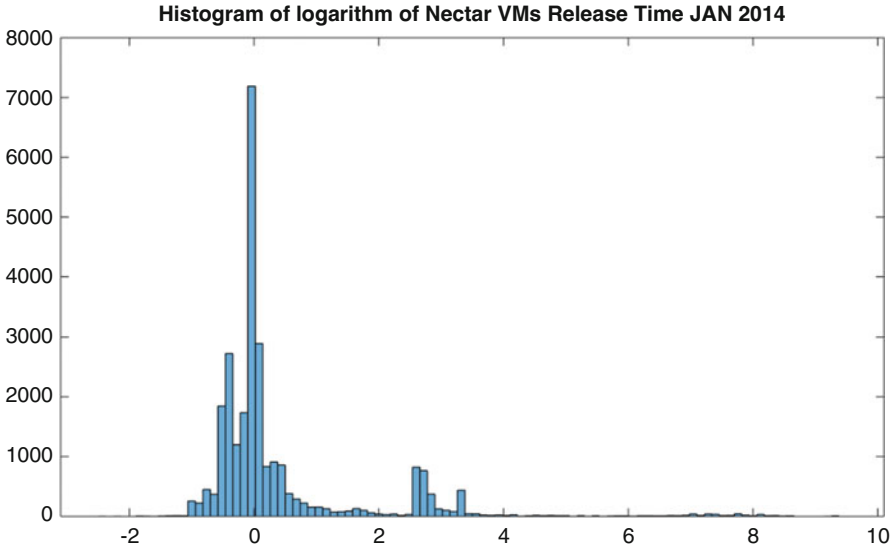

Fig. 3.3 Histogram of logarithm of release time (second) of VMs created in Nectar Cloud in January 2014

- To the best of our level of understanding of such existing DVMC algorithms as [8–23, 27], including *RTDVMC* [26] are developed based on the fundamental underlying assumption that optimal energy efficiency is achievable at maximum PM resource utilization. The basis of such claim against these existing DVMC algorithms is that these algorithms use legacy PMs, such as HP ProLiant ML110 G4 [28] and HP ProLiant ML110 G5 [29] for performance validation. From Fig. 3.4, we can see that for those legacy PMs the energy efficiency (i.e., Ratio of Power to Throughput) is as high as its incurred utilization. In stark contrast, for modern highly energy proportional PMs, such as Dell PowerEdgeR940 (Intel Xeon Platinum 8180, 112 cores $\rightarrow$ 25,000 MHz, 384 GB) [30], HP ProLiant DL560 Gen10 (Intel Xeon Platinum 8180, 112 cores $\rightarrow$ 25,000 MHz, 384 GB) [31], and HP ProLiant ML350 Gen10 (Intel Xeon Platinum 8180, 28 cores $\rightarrow$ 25,000 MHz, 192 GB) [32], energy efficiency rather drops beyond 70% utilization intervals, as delineated in Fig. 3.4. The underlying reason is that, while the throughput increases uniformly with the increase of load or utilization (Fig. 3.5), the power consumption of modern PMs beyond 70% utilization level rises such drastically (Fig. 3.6) that it exceeds the respective linear increase of throughput. Consequently, the ratio of throughput to power consumption, translated as energy efficiency, drops [33]. As such, several researchers [34, 35] argue that consolidation towards maximum increase of PM resource utilization does not feature the optimal minimization of CDC energy consumption with new generation of highly energy proportional PMs. Based on our literature study [8–23, 27], no DVMC algorithm including *RTDVMC* is found to address this issue.

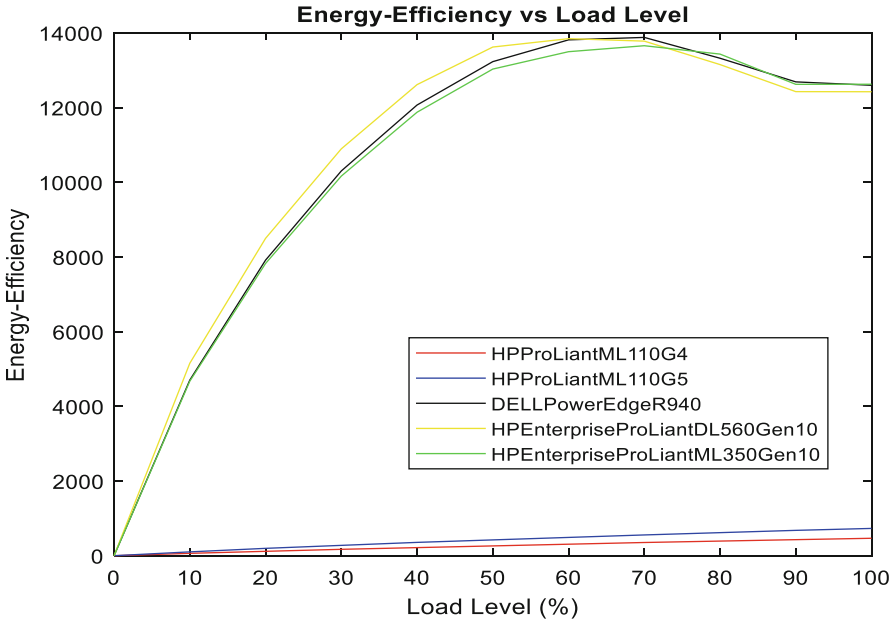

Fig. 3.4 Change of energy efficiency with varying load level for various PMs

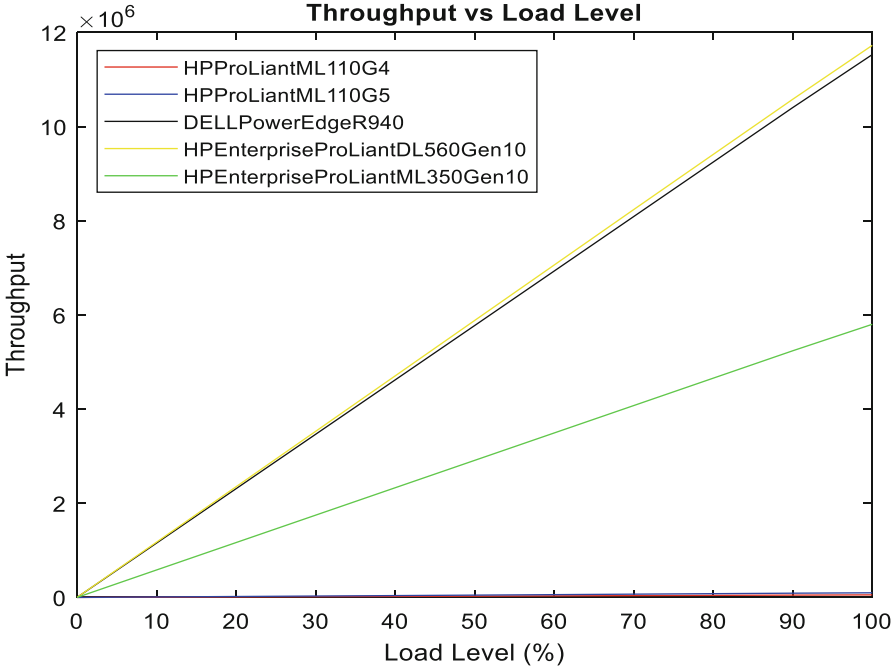

Fig. 3.5 Change of throughput with varying load level for various PMs

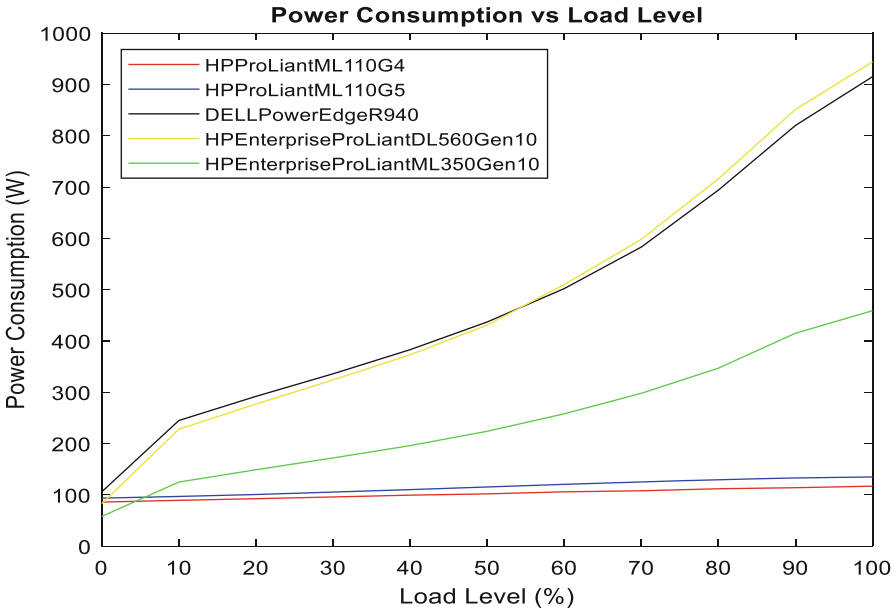

Fig. 3.6 Change of power consumption with varying load level for various PMs

- Preventing Quality of Service (QoS) degradation in CDC due to excessive VM migration is as much important as it is to minimize CDC energy consumption. However, higher energy consumption minimization by *RTDVMC* comes at a cost of excessive increase of VM migration.
- While, performance of *RTDVMC* has strong correlation with the value of VMRT, instead of VMRT values of real Cloud workload, only simulated VMRT values have been used. The complex behavior and interaction of VMs have impact on VMRT values, which cannot be reflected in simulated VMRT values.
- *RTDVMC* consolidates VMs primarily on the value of VMRT with an assumption that VMRT would be precisely known in advance, whereas, in reality VMRT would not be strictly accurate in many scenarios.

With respect to above-mentioned challenges, our contributions in this paper are briefly outlined in the following.

- A novel heuristic DVMC algorithm, namely, *Stochastic Release Time Based DVMC* (*SRTDVMC*) algorithm, has been proposed, which is robust to uphold optimal performance in terms of minimizing energy consumption regardless of the potential change in underlying PMs' energy efficiency characteristics.
- One crucial goal of experiment is to predict behavior of an algorithm in the real world [36]. For performance evaluation, VMRT values extracted from real Cloud, namely, Nectar Cloud, has been used, which assists to obtain a stronger performance prediction under real Cloud scenarios.

- While minimization of VM migration reduces network overhead and subsequent energy consumption by networking equipment, it is intrinsic in VM consolidation. *SRTDVMC* is multi-objective, which concomitantly addresses two confronting goals: minimizing energy consumption and minimizing VM migration.
- *SRTDVMC* eliminates the restriction of strictly accurate VMRT through the conversion of VMRT into respective Stochastic VMRT (SVMRT).

The organization of the rest of the paper is as follows. In Sect. 3.2, related literature is highlighted. Stochastic Release Time and various notations used in the algorithm are elucidated in Sect. 3.3. Next, in Sect. 3.4, we have brought forth the proposed algorithm, followed by elucidation of core components and characteristics of *SRTDVMC*. In Sects. 3.5 and 3.6, the experimental setup and performance evaluation of the proposed algorithm would be articulated. In Sect. 3.7, we have elucidated our critical observations extracted from empirical evaluations. Finally, in Sect. 3.8, we have summarized our research with future research directions and motivation.

3.2 Related Work

In depth review and classification of VMC algorithms have been broadly discussed in [37]. VMC is a NP-hard problem and can be broadly classified into two groups: Static VM Consolidation (SVMC) [38–40] and DVMC. SVMC algorithm provides the solution of energy-efficient initial VM placement [38]. However, as time progresses, the initial VM placement solution loses the efficiency with the change in resource demand and the resource availability. In case of increase in resource demand, VM(s) might be required to be migrated out into other suitable PM(s), while the decrease in resource demand widens up the room to host additional VM(s) and, hence, presents the opportunity to minimize the CDC energy consumption further by consolidating more VMs in in lesser number of PMs than before. Contrast to SVMC, dynamic nature of DVMC algorithm contributes in further energy consumption minimization, as it captures the opportunity arisen from varying resource demand and resource availability and iteratively consolidate VMs whenever consolidation opportunity arises.

DVMC algorithms can be broadly classified into two groups: Centralized DVMC [41–43] and Distributed DVMC [27, 44]. Major DVMC algorithms found in the literature are centralized DVMC and only a few Distributed DVMC [27, 44] has been proposed. In [27], authors have presented their distributed DVMC algorithm for a P2P network oriented CDC. According to [27, 44], the growing number of PMs becomes a bottleneck for CDVMC at the time of selecting a destination for any migrating VM, since the asymptotic time complexity of the centralized DVMC algorithm is proportional to the number of PMs in the CDC, whereas the number of potential PMs to choose from for a migrating VM is relatively small in distributed

DVMC. Thus, distributed DVMC is more scalable for CDC with huge number of PMs. However, distributed DVMC has message passing overhead, as every PM must update its present resource availability to all of its neighbors. Message passing increases network overhead, decreases network throughput, and increases network-related energy consumption. Both centralized and distributed DVMC algorithms can be classified into two groups: Static Threshold-Based DVMC algorithm [8, 11, 26] and Adaptive Threshold-Based DVMC algorithm [12, 23]. The later increases VM migration. Our objective is to minimize both energy consumption and VM migration. Hence, in this paper, we have opted to Static Threshold-based DVMC approach.

Elucidated earlier in Sect. 3.1, performance of existing VMC algorithms lacks in terms of minimizing CDC energy consumption with the rise of highly energy proportional PMs. PEAS [39] and EPACT [45] as SVMC algorithms address this issue. However, on account of being an SVMC algorithm, those algorithms do not continue to dynamically consolidate VMs whenever opportunity arises, and hence, the solution would lose optimality over time. Based on our literature study [8–23, 27], no existing DVMC algorithm is designed considering energy-efficiency characteristics of highly energy proportional state-of-the-art PMs in their bedrocks. As such, we have brought forth *SRTDVMC* which takes the dynamics of modern PMs' energy-efficiency characteristics with respect to varying load level into account and limits both VM migration and CDC energy consumption.

In our proposed *SRTDVMC* algorithm, stochastic release time is one of the key distinctive features used in VMC decision process. To ensure easy understanding of our proposed *SRTDVMC* algorithm, we have first explained the key terms used in the proposed algorithm in following section.

3.3 Modelling Stochastic VM Release Time, Notations Used and Important Concepts

3.3.1 Modelling Stochastic VM Release Time, PM Release Time and Notations Used

Workload finishing time or lifetime of a VM is referred to as VMRT. Prior receiving any service, negotiation of service related conditions including service expiry date followed by an acceptance of the contract takes place between Cloud Service Providers (CSPs) and CSUs, interpreted as Service Level Agreement (SLA). For many VMs, VMRT is equal to the contract of service period between the respective CSU (i.e., VM owner) and the CSP as agreed during SLA. Before the contract is expired, both CSUs and CSPs might agree/disagree to renew, extend or early termination of the contract and respective VMRT would be updated accordingly. Some web applications hosted in CDC remains unremoved for a very long period. Estimated VMRT of such VMs would be large values referring to the time when the

respective contract of service between CSU and CSP would expire as agreed prior the service during SLA. If renewal or early termination of contract of service takes place, VMRT would be readjusted accordingly.

For some applications, VMRT corresponds to QoS and potential resource demand. Resource demand may change with the variation in number of users, causing creation of additional VMs and later deletion of such VMs. At the time of SLA, a SAAS provider/PAAS user must consider the potential number of application users and mention it to PAAS provider so that by taking the potential number of end users and corresponding resource demand into account a certain standard of QoS can be uphold. Pattern of changing resource demand over time derived from past data can also be utilized to recognize the change of resource demand in future [46]. Considering the change in resource demand over time and demanded QoS, PAAS provider/IAAS user can estimate resource/VM release time, which would be proffered to IAAS provider.

In many cases, the prior estimated VMRT might not turn out as strictly accurate in future. Hence, to reflect the reality further closely, we propose to embody a stochastic version of VMRT, referred to as Stochastic VM Release Time (SVMRT) in *SRTDVMC*. SVMRT can be calculated from (3.1). In Table 3.1, we have articulated the meaning of the notations used in this paper.

$$S_{V_j} = (1 + \alpha \cdot Y) \cdot T_{V_j} \tag{3.1}$$

Apart from *VMRT*, two more crucial terms noteworthy explaining is *PM Release Time* (*PMRT*) and Stochastic *PMRT* (*SPMRT*). *PMRT* refers to the time when a PM can be either shut down or put into a sleep state that would consume no energy, or lower amount of energy compared to its active state. A PM can be shut down or put into sleep state, if it has either no VM hosted on it, or none of its hosted VMs is in the active state. Since *SVMRT* refers to the maximum time until which the VM would be in the active state, hence *SPMRT* of a PM P_i denoted by S_{P_i} refers to the maximum *SVMRT* value among all the *VMRT* values of VMs that are hosted in that PM, as articulated in (3.2).

$$S_{P_i} = \max\left(S_{V^i}\right) \tag{3.2}$$

3.3.2 Modelling Resource Utilization and Constraints

DVMC algorithm migrates VMs from one PM to another PM, so that VMs would be placed in minimum number of PMs and thus increase resource utilization and energy efficiency. VMs hosted in a PM utilize the resources of that PM. Therefore, resource utilization of a PM corresponds to the total resource demand by all the VMs hosted in that PM. Let U_i^k denote utilization of Resource type, R_k of PM, P_i. Hence, the equation for calculating U_i^k [47] is as follows:

Table 3.1 Notations used

Notations	Meaning
$\|\|$	Cardinality of a set
$P = \{P_i\}_{\|P\|}$	The set of $\|P\|$ PMs
$V = \{V_j\}_{\|V\|}$	The set of $\|V\|$ VMs
$V^i = \left\{V_j^i\right\}_{\|V^i\|}$	Set of VMs hosted in PM, P_i
T_{V_j}	*VMRT* of VM, V_j
$T_{V_j^i}$	*VMRT* of VM, V_j^i
$T_{V^i} = \left\{T_{V_j^i}\right\}$	The set of *VMRT* of VMs hosted in PM, P_i
Y	Random variable ranging $[-1, +1]$
α	Maximum deviation of *VMRT*
S_{V_j}	S*VMRT* of VM, V_j
$S_{V_j^i}$	S*VMRT* of VM, V_j hosted in PM, P_i
$S_{V^i} = \left\{S_{V_j^i}\right\}$	The set of S*VMRT* of VMs hosted in PM, P_i
S_{P_i}	*SPMRT* of PM, P_i. $S_{P_i} = \max\left(S_{V^i}\right)$
$x = \{x_{i,j}\}_{\|P\| \times \|V\|}$	Placement matrix
$R = \{R_k\}_{\|R\|}$	The set of different types of resources
D_j^k	Demand of resource, R_k by VM, V_j
U_i^k	Utilization of resource, R_k of PM, P_i
C_i^k	Capacity of PM, P_i in terms of resource, R_k
θ_{MAX}	Maximum threshold
$\text{OP} = \{P_o\}_{\|\text{OP}\|}$	The set of *O-UPMs*
$V^o = \left\{V_q^o\right\}_{\|V^o\|}$	The set of VMs hosted in an *O-UPM*, P_o
D_q^k	Demand of resource, R_k by VM, V_q^o
U_o^k	Utilization of resource, R_k of PM, P_o
C_o^k	Capacity of PM, P_o in terms of resource, R_k
$\text{NOP} = \{P_x\}_{\|\text{NOP}\|}$	The set of non-*over-utilized PMs*, which are neither *O-UPMs* nor in sleep mode or switched off
$\text{SP} = \{P_s\}_{\|\text{SP}\|}$	The set of PMs that are in sleep mode or switched off

P_d	Destination PM
$vmsToMigrate = \{V_l\}_{\lvert vmsToMigrate\rvert}$	The set VMs to migrate out into new PMs
$destinationPMs = \{P_d\}_{\lvert destinationPMs\rvert}$	The set of new destination PMs for migrating VMs from source *O-UPMs*
$candidateSources = \{P_c\}_{\lvert candidateSources\rvert}$	The set of PMs from which a PM would be selected as *U-UPM*
$V^c = \left\{V_n^c\right\}_{\lvert V^c\rvert}$	The set of VMs hosted in PM, P_c
$candidateDestinations = \{P_m\}_{\lvert candidateDestinations\rvert}$	The set of PMs from which a PM would be selected to host migrating VM of an *U-UPM*
$Destinations$	The set of new destination PMs for migrating VMs of source *U-UPMs*
E_i	Energy consumption by PM, P_i
E_{CDC}	Energy consumption by the CDC
$\overline{E}_{\text{CDC}}$	Mean energy consumption by the CDC
f_1	First objective function of *SRTDVMC*
f_2	Second objective function of *SRTDVMC*
$Nectar = \{Nectar_{NT}\}_{\lvert 3\rvert} = \{\text{Nov 2013, Dec 2013, Jan 2014}\}$	Set of different months of Nectar Cloud
$PLab = \{PLab_{PL}\}_{\lvert 4\rvert} = \{\text{6 March, 9 March, 9 April, 20 April}\}$	Set of different days of PlanetLab Cloud
$\overline{E}^S = \left\{\overline{E}^S_{NT,PL}\right\}_{\lvert Nectar\rvert\cdot\lvert PLab\rvert}$	Set of mean energy consumption values for *SRTDVMC* algorithm under different workload combinations of Nectar and PlanetLab
$\overline{E}^R = \left\{\overline{E}^R_{NT,PL}\right\}_{\lvert Nectar\rvert\cdot\lvert PLab\rvert}$	Set of mean energy consumption values for *RTDVMC* algorithm under different workload combinations of Nectar and PlanetLab
$np = \{1, 2, \ldots, \lvert Nectar\rvert \cdot \lvert PLab\rvert\}$	Index to denote *np*th smallest element of $\overline{E}^S$, $\overline{E}^R$, $\overline{\psi}^S$ and $\overline{\psi}^R$
a_{np}	Weights related to Shapiro-Wilk Normality Test
$\overline{E}^S_{(np)}$	*np*th smallest element of $\overline{E}^S$
$\overline{E}^R_{(np)}$	*np*th smallest element of $\overline{E}^R$

(continued)

Table 3.1 (continued)

Notations	Meaning
$\overline{\overline{E}}^{S}$	Mean of $\overline{E}^{S}$
$\overline{\overline{E}}^{R}$	Mean of $\overline{E}^{R}$
ψ	Total number of VM migration
$\overline{\psi}$	Mean total number of VM migration
$\overline{\psi}^{S} = \left\{ \overline{\psi}^{S}_{NT,PL} \right\}_{\lvert Nectar \rvert \cdot \lvert PLab \rvert}$	Set of mean number of VM migration for *SRTDVMC* algorithm under different workload of Nectar and PlanetLab
$\overline{\psi}^{R} = \left\{ \overline{\psi}^{R}_{NT,PL} \right\}_{\lvert Nectar \rvert \cdot \lvert PLab \rvert}$	Set of mean number of VM migration for *RTDVMC* algorithm under different workload of Nectar and PlanetLab
$X^{\overline{E}} = \left\{ X^{\overline{E}}_{NT,PL} \right\}_{\lvert Nectar \rvert \times \lvert PLab \rvert}$	Minimization of $\overline{E}_{\mathrm{CDC}}$ by *SRTDVMC* compared to *RTDVMC* under different workload combination of Nectar and PlanetLab
$X^{\overline{\psi}} = \left\{ X^{\overline{\psi}}_{NT,PL} \right\}_{\lvert Nectar \rvert \times \lvert PLab \rvert}$	Minimization of $\overline{\psi}$ by *SRTDVMC* compared to *RTDVMC* under different workload combination of Nectar and PlanetLab
$\mathrm{SW}^{S}_{\overline{E}}$	Test statistics for the S-W normality test with $\overline{E}^{S}$
$\mathrm{SW}^{R}_{\overline{E}}$	Test statistics for the S-W normality test with $\overline{E}^{R}$
$\mathrm{SW}^{S}_{\overline{\psi}}$	Test statistics for the S-W normality test with $\overline{\psi}^{S}$
$\mathrm{SW}^{R}_{\overline{\psi}}$	Test statistics for the S-W normality test with $\overline{\psi}^{R}$
$t_{\overline{X}^{\overline{E}}}$	Test statistic for the *t-test* with data samples of $\overline{X}^{\overline{E}}$
$t_{\overline{X}^{\overline{\psi}}}$	Test statistic for the *t-test* with data samples of $\overline{X}^{\overline{\psi}}$
$\hat{\sigma}_{\overline{X}^{\overline{E}}}$	Standard deviation of $\overline{X}^{\overline{E}}$
$\hat{\sigma}_{\overline{X}^{\overline{\psi}}}$	Standard deviation of $\overline{X}^{\overline{\psi}}$
$\overline{X}^{\overline{E}}$	Mean of $X^{\overline{E}}$
$\overline{X}^{\overline{\psi}}$	Mean of $X^{\overline{\psi}}$

$$U_i^k = \sum_{j=1}^{|V|} D_j^k \cdot x_{i,j} \tag{3.3}$$

where D_j^k denotes Demand of Resource type, R_k by VM, V_j, $x_{i,j}$ denotes the element of placement matrix, x and value of $x_{i,j}$ is determined as follows:

$$x_{i,j} = \begin{cases} 1, \text{ if } V_j \text{ is placed in } P_i \\ 0, \text{ otherwise} \end{cases} \tag{3.4}$$

Since, Resource Capacity of a PM, P_i is fixed and it cannot provide additional resources to its hosted VMs than its capacity, Therefore, a VM, V_j can only be placed in a PM, P_i if the amount of available resources in P_i is adequate to meet the resource demand of V_j. Hence, V_j can only be placed in P_i if the following equation is satisfied [47]:

$$C_i^k - U_i^k \geq D_j^k \tag{3.5}$$

In other words, a PM, P_i cannot host a VM, V_j if the available resource of P_i is lesser than the resource demand of V_j. We denote such constraint presented in (3.5) as *Resource Constraint* (*RC*).

At times, workload of VMs could rise very high resulting in steep resource utilization of the hosting PM. We denote such PM with heavy resource utilization as *Over-utilized PM* (*O-UPM*) and use a threshold, referred to as *Maximum Threshold,* $\theta_{\max}$ to distinguish whether a PM is *Over-utilized* or not. Let us denote *OP* (3.6) as a set of *O-UPMs*.

$$\text{OP} = \left\{ P_i \mid U_i^k \geq \theta_{\max} \text{ for any } R_k \in R \text{ and } 1 \leq i \leq |P| \right\} \tag{3.6}$$

If VM(s) were not migrated out of an *O-UPM, then* SLA violation would unfold. In order to avoid causing SLA violation, during destination PM selection for a migrating VM, it is essential to ensure that hosting the migrating VM would not turn the destination into an *O-UPM*, which we have modelled through the following equation:

$$D_j^k + U_i^k < \theta_{\max} \tag{3.7}$$

We refer such constraint presented in (3.7) as *Maximum Utilization Threshold Constraint* (*MUTC*).

3.3.3 Modelling Energy Consumption

Most of the existing VM consolidation algorithms have mentioned that energy consumption of a PM is primarily dominated by its CPU utilization [8, 47]. Hence, our energy consumption model is a function of CPU utilization (3.8), where, E_i denotes the energy consumption by PM, P_i.

$$E_i = f\left(U_i^{\text{CPU}}\right) \tag{3.8}$$

Based on (3.8), we can determine the total energy consumption of the CDC through (3.9), where E_{CDC} denotes the total energy consumption of the CDC.

$$E_{\text{CDC}} = \sum_{i=1}^{|P|} E_i \tag{3.9}$$

In order to relate closely to the real energy consumption by PMs, we have opted to draw energy consumption benchmark results of three different types of PMs presented in Table 3.2.

3.3.4 Objective Functions

DVMC algorithms aim to minimize CDC energy consumption through migrating VM(s) out of lower utilized PMs and placing those VM(s) in relatively higher utilized PM(s). As such, the first objective function, f_1 of *SRTDVMC* has been characterized through (3.10).

$$f_1 = \min\left(\overline{E}_{\text{CDC}}\right) \tag{3.10}$$

One downfall of DVMC is that energy consumption minimization through VM consolidation cannot be achieved without VM migration, which itself deteriorates

Table 3.2 Energy consumption values of contemporary servers at different load level

	Energy consumption (kW) at different percentage of load level										
	Sleep	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Dell Power Edge R940	106	245	292	336	383	437	502	583	694	820	915
HP ProLiant DL560 Gen10	82.9	228	277	324	373	431	510	598	716	851	944
HP ProLiant ML350 Gen10	58.1	125	149	172	196	224	258	298	347	415	459

QoS as well as raises network overhead leading towards increased SLA violation and increased energy consumption by networking equipment of CDC. Therefore, restricting VM migration in CDC is no less important than lowering CDC energy consumption. As such, (3.11) characterizes the second objective function, f_2 of *SRTDVMC*.

$$f_2 = \min\left(\overline{\psi}\right) \tag{3.11}$$

It is worthwhile to note that f_1 and f_2 are two confronting objective functions. Value of f_1 can only be minimized through migrating VM(s) from lower utilized PM(s) to relatively higher utilized PMs, which increases ψ, whereas, increased ψ negatively affects f_2. Hence, it is more challenging to design a DVMC algorithm, which can show better performance in terms of both f_1 and f_2. In the following section, our proposed multi-objective heuristic DVMC algorithm, namely, *SRTDVMC* has been presented, which aims to optimize f_1 and f_2.

3.4 Proposed Solution

At the outset of the paper, in Sect. 3.1, we have elucidated the limitations of *RTDVMC*. To address those limitations, we have proposed our *SRTDVMC* algorithm presented in Algorithm 3.1. The meaning of notations used in *SRTDVMC* algorithm has been presented earlier in Table 3.1.

Algorithm 3.1: The SRTDVMC algorithm

Input: P, V, R, SP
Output: VM Placement
The first phase: O-UPMs
1: **for** each P_i in P **do**
2: **if** (3.6) is satisfied **then**
3: $OP \leftarrow \{P_i \cup OP\}$
4: **end if**
5: **end for** // end of for loop from Line 1 to 5
6: **for** each P_o in OP **do**
7: *migratingVMs* ← Invoke the *VSO* algorithm with P_o
8: *vmsToMigrate* ← {*migratingVMs* ∪ *vmsToMigrate*}
9: **end for** // end of for loop from Line 6 to 9
10: Sort *vmsToMigrate* in the order of decreasing *SVMRT*
11: $NOP \leftarrow \{P - OP - SP\}$
12: **for** each V_l in sorted *vmsToMigrate* **do**
13: $P_d \leftarrow$ Invoke the *DPSVO* algorithm with V_l and NOP
14: *destinationPMs* ← {P_d ∪ *destinationPMs*}
15: **if** P_d is in SP **then**
16: $SP \leftarrow \{SP - P_d\}$
17: **for** each R_k in R **do**

```
18:              U_d^k ← U_d^k + D_q^k
19:              C_d^k ← C_d^k - D_q^k
20:          end for    // end of for loop from Line 17 to 20
21:          NOP ← {P_d ∪ NOP}
22:      end if         // end of if at Line 15
23:  end for           // end of for loop from Line 12 to 23
     The second phase: U-UPMs
24:  candidateSources ← {P - OP - SP - destinationPMs}
25:  candidateDestinations ← {P - OP - SP}
26:  Sort candidateSources in the order of increasing SPMRT
27:  for each P_c in sorted candidateSources do
28:      candidateDestinations ← {candidateDestinations - P_c}
29:      destinations ← Invoke the DPSVU algorithm with P_c
                and candidateDestinations
30:      candidateSources ← {candidateSources - destinations}
31:  end for           // end of for loop from Line 27 to 31
```

3.4.1 Two Phases of SRTDVMC Algorithm

In this section, we have explained our proposed *SRTDVMC* algorithm. Resource demand of VMs may change over time, resulting variation of resource utilization in host PMs. When resource demand of hosted VM(s) grow higher over time, the hosting PM might fall short of adequate resources to prevent potential performance degradation, translated as SLA violation. Again, hosted VMs resource demand may decline over time, bringing forth a gap in the hosting PM to accommodate additional VMs from other low utilized PMs, leading towards potential opportunity to reduce the number of active PMs and subsequent reduction of CDC energy consumption. Based on this principal, *SRTDVMC* algorithm works in two phases – the first phase and the second phase.

In *the first phase*, VMs from *O-UPMs* are migrated out to control SLA violations (Line 1–23 of Algorithm 3.1) and in *the second phase*, VMs from *U-UPMs* are migrated out to consolidate in lesser number of active PMs (Line 24–31 of Algorithm 3.1). In the following sections, we have comprehensibly discussed each of these phases and its components.

3.4.1.1 The First Phase O-UPMs

As expressed in (3.6), for any resource type (i.e., CPU, RAM and Bandwidth), if a PM's resource utilization is found as equal or greater than $\theta_{\max}$, then the PM is denoted as *O-UPM* (Line 1–5 of Algorithm 3.1). Next, VM(s) are selected from all *O-UPMs* to migrate out into new destination PMs (Line 7 of Algorithm 3.1). The *VSO* algorithm (Algorithm 3.2) proffers to the VM(s), which are attempted to migrate out from an *O-UPM*. In the following section, we have explained the *VSO* algorithm.

- **VM Selection from *O-UPMs***

VMs of an *O-UPM* are first sorted in decreasing order of VMRT (Line 1 of Algorithm 3.2). The first VM from the sorted list of VMs is first selected and checked whether the respective PM's utilization drops lower than $\theta_{\max}$. If yes, then the respective PM is no more *O-UPM* and VM selection process stops (Line 2–3 of Algorithm 3.2). Otherwise, the second VM from the sorted list of VMs is selected and then the third VM and so forth, until the PM's utilization is decreased below $\theta_{\max}$ (Line 3–10 of Algorithm 3.2). The rationale of selecting VM(s) with largest VMRT is to minimize the active duration of source *O-UPM*, which might aid in minimization of energy consumption.

Algorithm 3.2: The VM Selection from *O-UPM* (*VSO*) Algorithm

Input: The *O-UPM*, P_o.
Output: List of VMs to be migrated out from the given *O-UPM*
1: Sort V^o, the set of VMs of P_o in order of decreasing *VMRT*
2: $q \leftarrow 1$
3: **while** $q \leq |V^o|$ and $\theta_{MAX} < U_o{}^k$ **for** any R_k in R **do**
4: $migratingVMs \leftarrow \{V_q{}^o \cup migratingVMs\}$
5: **for** each R_k in R **do**
6: $U_o{}^k \leftarrow U_o{}^k$ - $D_q{}^k$
7: $C_o{}^k \leftarrow C_o{}^k + D_q{}^k$
8: **end for**
9: $q \leftarrow q + 1$
10: **end while**
11: **return** *migratingVMs*

- **Destination PM Selection for Migrating VMs from *O-UPMs***

SRTDVMC algorithm develops a set, denoted as *vmsToMigrate*, comprised of all migrating VMs from *O-UPMs* (Line 6–9 of Algorithm 3.1), as these VMs of *vmsToMigrate* are sorted in decreasing order of VMRT (Line 10 of Algorithm 3.1). The migrating VMs are attempted to host in PMs, which are neither *O-UPMs*, nor in sleep state or turned off state. Such set of PMs, which are not *O-UPMs* and not in either turned off or sleep state is referred to as *NOP* (Line 11 of Algorithm 3.1). *SRTDVMC* algorithm then keeps invoking *DPSVO* algorithm, presented as Algorithm 3.3 to determine the destination PM for each of the migrating VMs of *vmsToMigrate* starting from the largest VM to the smallest VM in terms of VMRT (Line 12–23 of Algorithm 3.1).

Algorithm 3.3: The Destination PM Selection for VM of *O-UPM* (*DPSVO*) Algorithm

Input: The VM V_j to be migrated out from an *O-UPM*
Input:*NOP*
Output: The new destination PM for the given migrating VM
1: Sort *NOP* in the order of increasing *SPMRT*
2: **for** each P_x in sorted *NOP* **do**
3: suitable $\leftarrow$ Invoke the *PST* algorithm with P_x and V_j

```
4:      if suitable is true then
5:          return P_x
6:      end if
7:  end for
8:  return most energy-efficient P_s, which satisfies (3.5) and (3.7)
```

Algorithm 3.4: The PM Suitability Test (*PST*) Algorithm

Input: *PM*, *VM*
Output: A decision whether the given *PM* can host the given *VM*

```
1:  if RC (3.5) and MUTC (3.7) are satisfied for all R_k in R then
2:      return true
3:  end if
4:  return false
```

In order to determine a PM from *NOP* as destination host for a migrating VM of *vmsToMigrate*, the *DPSVO* algorithm first sorts the PMs of *NOP* in increasing order of *SPMRT* (Line 1 of Algorithm 3.3). The smallest PM in terms of *SPMRT* from the sorted *NOP* is first checked whether it is suitable to accommodate the migrating VM or not (Line 2 of Algorithm 3.3). The *PST* algorithm presented in Algorithm 3.4 is invoked to check the suitability of a PM as a potential destination PM (Line 3 of Algorithm 3.3). A PM is considered suitable, if RC (3.5) and MUTC (3.7) constraints are not violated (Line 1–4 of Algorithm 3.4). If that PM is found as suitable as per *PST* algorithm, then it is selected as the new destination PM for the migrating VM and the destination PM selection process ends (Line 4–6 of Algorithm 3.3). In case the PM is found as unsuitable, suitability of the second smallest PM in terms of *SPMRT* from the sorted *NOP* is checked and then the third smallest PM and so forth until a suitable PM is found (Line 2–7 of Algorithm 3.3). If no PM from *NOP* can accommodate that particular migrating VM, then the most energy-efficient and suitable PM from the set of PMs, which are in either sleep or turned-off state, referred to as *SP* is awoke and selected as destination PM (Line 8 of Algorithm 3.3). If the destination PM is selected from *SP*, then that PM is removed from the set *SP*, since it is no more in sleep or turned-off state and its utilization and capacity values across all resource types are adjusted (Line 15–20 of Algorithm 3.1). Furthermore, the PM is added in the set *NOP*, so that it can be considered as a potential destination PM for following migrating VMs of *vmsToMigrate* (Line 21–22 of Algorithm 3.1).

3.4.1.2 The Second Phase U-UPMs

Under-Utilized PMs (*U-UPMs*) refers to the set of those PMs, which had not been determined as *O-UPMs* in the first phase of *SRTDVMC* and which do not belong to *SP*. After determining the destination PMs for VMs of *O-UPMs*, the second phase of *SRTDVMC* is commenced when VMs from *U-UPMs* are migrated out, followed by strategic destination PM selection for those migrating VMs with an aim to lower

the number of active PMs so that CDC energy consumption can be minimized (Line 24–31 of Algorithm 3.1).

- **Source PM Selection from *O-UPMs***

In the second phase, *SRTDVMC* first rounds up a set of *U-UPMs*, namely, *candidateSources* – the set representing potential source *U-UPM*(s) from which VM(s) would be attempted to migrate out. The set of PMs, which were identified as *O-UPMs* in the first phase, referred to as *OP* along with the set of PMs, which hosted migrating VMs of *O-UPMs* are excluded from *candidateSources* to avoid repeated handling of PMs from *OP* and to inhibit re-migration of those VMs, which had been migrated out once in the first phase (Line 24 of Algorithm 3.1). The PMs of *candidateSources* are then sorted in the increasing order of *SPMRT* (Line 26 of Algorithm 3.1). All PMs of sorted *candidateSources* starting from the smallest PM to the largest PM in terms of *SPMRT* is sequentially selected as source *U-UPM*. However, the set of *U-UPMs* denoted by *destinations*, which are determined as the new destination PM(s) for migrating VM(s) from a source *U-UPM* is excluded from *candidateSources* and hence, those new destination *U-UPMs* cannot become source *U-UPM*, which prevents repeated migration of same VMs (Line 30 of Algorithm 3.1).

- **Migrating VM and Destination PM Selection**

SRTDVMC algorithm invokes the *DPSVU* algorithm (Algorithm 3.5) to select migrating VMs from *U-UPMs* and corresponding new destination PMs. Once a *U-UPM* from sorted *candidateSources* is selected as source *U-UPM*, the hosted VMs in that source *U-UPM* is sorted in decreasing order of VMRT (Line 1 of Algorithm 3.5). The VMs starting from the largest to the smallest in terms of VMRT are attempted to migrate out (Line 2 of Algorithm 3.5). The reason of selecting VMs in descending order of VMRT is that migrating out the largest VM can reduce the *SPMRT* of the source PM leading towards energy consumption minimization. If for any VM, a suitable new destination *U-UPM* cannot be found, the migrating VM(s) selection from a source *U-UPM* terminates (Line 18–20 inside of Line 2–21 from Algorithm 3.5). In the following, we have discussed the process of determining the new destination PM for such migrating VM.

Algorithm 3.5: The Destination PMs Selection for VMs of *U-UPMs* (*DPSVU*) Algorithm

Input: An *U-UPM*, P_c from the set of *candidateSources*
Input:*candidateDestinations*
Output: List of new destination PMs to host migrating VMs
1: Sort V^c, the set of VMs of P_c in order of decreasing *VMRT*
2: **for** each ${V_n}^c$ in sorted V^c **do**
3: $P_d \leftarrow$ **null**
4: Sort *candidateDestinations* in order of increasing *SPMRT*
5: **for** each P_m in sorted *candidateDestinations* **do**
6: suitable ← Invoke the *PST* algorithm with P_m and ${V_n}^c$
7: **if** suitable is **true**

```
8:              energyDrop ← Energy drop in P_c without V_n^c
9:              energyRise ← Energy rise of P_m for hosting V_n^c
10:             netEnergyGain ← EnergyDrop - EnergyRise
11:             if NetEnergyGain > 0
12:                 P_d ← P_m
13:                 hostList ← {P_d ∪ hostList}
14:                 break loop
15:             end if
16:         end if
17:     end for
18:     if P_d is null then
19:       break loop
20:     end if
21: end for
22: return hostList
```

In order to select the destination PM for a migrating VM of *U-UPM*, *SRT-DVMC* algorithm first creates a set of potential destination PMs, referred to as *candidateDestinations*. The PMs of *SP, OP* and the source *U-UPMs* hosting the migrating VMs are excluded from *candidateDestinations*, since a source PM cannot be the new destination PM of its own VMs and to avoid increasing the likelihood of turning the PMs from *OP* into *O-UPMs* again (Line 25, 27 and 28 of Algorithm 3.1). The *DPSVU* algorithm (Algorithm 3.5) is then invoked to select the destination PM from *candidateDestinations* (Line 29 of Algorithm 3.1). The PMs of *candidateDestinations* are first sorted in increasing order of *SPMRT* (Line 4 of Algorithm 3.5) and then the suitability of these PMs from sorted *candidateDestinations* are sequentially checked starting from the smallest to the largest PM in terms of *SPMRT* (Line 5–6 of Algorithm 3.5). If a PM is found as suitable satisfying both RC (3.5) and MUTC (3.7) constraints as per *PST* Algorithm (Algorithm 3.4), then net energy gain for the potential VM migration is estimated from the difference between reduced energy consumption of source *U-UPM* and increased energy consumption of new destination *U-UPM*. If net energy gain is found as positive, then that PM is selected as the new destination PM (Line 7–17 of Algorithm 3.5). In the following section, we have discussed the characteristics of *SRTDVMC* algorithm.

3.4.2 Characteristics of Proposed Algorithm

SRTDVMC attempts to fit the largest VM in terms of VMRT from the smallest PM in terms of *SPMRT* into the next smallest possible PM. Such consolidation approach shortens the *SPMRT* of source PM without raising the *SPMRT* of destination PM, resulting into decreased energy consumption. Additionally, selecting the next smallest possible PM as destination PM ensures that the PM is accomplishing largest possible jobs before moving into sleep state or turned off state. Consequently, remaining workload for the existing active PMs becomes lower, which aids in

energy consumption minimization. Furthermore, lesser remaining workload for existing active PMs increases the likelihood that upcoming workload can be served by these active PMs without turning on PMs, which are in lower energy consumption state, for instance, sleep or turned off state. Hence, energy consumption minimization is complemented.

One critical aspect of *SRTDVMC* is that both rise of energy in potential destination PM (i.e., cost) and drop of energy in potential source PM (i.e., benefit) is checked prior any potential VM migration. VMs from *U-UPMs* are migrated only if the net energy gain (i.e., energy drop − energy rise) is positive, which limits the number of VM migrations and improves QoS without compromising energy efficiency. Hence, *SRTDVMC* can concurrently satisfy both objective functions (3.10) and (3.11). Furthermore, *SRTDVMC* smartly selects destination PMs ensuring that the increased energy consumption of potential destination *U-UPM* does not outweigh the reduced energy consumption of potential source *U-UPM*. It aids to uphold the energy efficiency of the solution regardless of the drastic rise of state-of-the-art PMs' energy consumption causing declined energy efficiency at utilization level beyond 70%. Thus, *SRTDVMC* encounters the lack of energy-efficiency issue in the presence of state-of-the-art PMs as experienced with existing DVMC algorithms. As a result, *SRTDVMC* is robust against underlying PMs' change of energy-efficient characteristics with varying load.

3.5 Performance Evaluation

Figures 3.1, 3.2, and 3.3 representing the diverse range of VMRT of Nectar Cloud reveal the heterogeneous nature of real Cloud workloads in terms of finishing time. To the best of our knowledge, none of the existing DVMC algorithms except *RTDVMC* is designed considering heterogeneous VMRT in their bedrock assuming all jobs finish at the same time, which is unrealistic. Consequently, these traditional DVMC algorithms cannot provide optimal solution for heterogeneous VMRT. As such, we have compared the performance of *SRTDVMC* with *RTDVMC*, since both are developed considering heterogeneous VMRT in their bedrocks.

3.5.1 Experimental Setup

Performance of *RTDVMC* [26] has been evaluated through CloudSim [24]. Since, performance of *SRTDVMC* has been compared with *RTDVMC*, therefore, we have modelled and simulated a cloud environment in CloudSim [24], which we have used to simulate *SRTDVMC* algorithm under different workload scenarios. For fair comparison, both algorithms have been simulated using same environment with respect to the characteristics of CDC, VM, PM and energy module. The simulated CDC consists of 800 heterogeneous PMs. Three different modern generation of

PMs, such as Dell PowerEdgeR940 (Intel Xeon Platinum 8180, 112 cores → 25,000 MHz, 384 GB) [30], HP ProLiant DL560 Gen10 (Intel Xeon Platinum 8180, 112 cores → 25,000 MHz, 384 GB) [31], and HP ProLiant ML350 Gen10 (Intel Xeon Platinum 8180, 28 cores → 25,000 MHz, 192 GB) [32] have been used. Each server is provided with 1 GB/s network bandwidth. The energy consumption characteristics of these servers with varying workload is articulated in Table 3.2.

The characteristics of different VM types match with the VMs used by *RTDVMC* and correspond to Amazon EC2 instance types [48]. However, the difference between the simulated VMs and Amazon EC2 instance types is that the simulated VMs are single-core, which is explained by the fact that the workload data used for the simulations come from single-core VMs. Since, the single-core is used, 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).

Lifetime of a VM V_j, aka VMRT of a VM V_j, denoted by T_{V_j} can be different from one VM to another (i.e., heterogeneous). For further accurate estimation of the performance of both *SRTDVMC* and *RTDVMC* algorithms under real Cloud scenario, T_{V_j} values are drawn from VMRT traces of a real Cloud, namely, Nectar Cloud. Nectar Cloud consists of over thousands of VMs across multiple data centers located in eight different cities of Australia [7]. For *SRTDVMC* algorithm, T_{V_j} is converted into *SVMRT*, S_{V_j} as per (3.1), using 0.05 as the value of α and a uniformly distributed random variable ranging $[-1, +1]$ as X. For further clarity, maximum deviation of T_{V_j} from S_{V_j} is $\pm 5\%$. At the outset, VMs are provided with the resources defined by the VM types. However, during the lifetime, VMs utilize less resources according to the workload data, widening opportunities for dynamic consolidation. The workload data also reflects traces of real Cloud workload traffic, originated as part of the CoMon project, a monitoring infrastructure for PlanetLab [49]. For both *RTDVMC* and *SRTDVMC*, upper utilization threshold, $\theta_{\max}$ is considered as 80%. With every workload scenario, a DVMC algorithm has been run twice to generate mean CDC energy consumption and mean total number of VM migration by that DVMC algorithm under such workload scenario. Each time, the simulation has been run until 24 h CloudSim simulation clock time.

3.5.2 Performance Metrics and Workload Data

3.5.2.1 Performance Metrics

SRTDVMC is a multi-objective DVMC algorithm, which aims to minimize the CDC energy consumption and VM migrations. Hence, performance of *SRTDVMC* and *RTDVMC* algorithms have been measured and compared in terms E_{CDC} and ψ. Expressed in (3.9), E_{CDC} is the sum of energy consumption of all the PMs, while each PM's energy consumption is derived from Table 3.2 according to its current

Table 3.3 Characteristics of PlanetLab Data (CPU Utilization)

Day	Number of VMs	Mean	St. dev.	Quartile 1 (%)	Quartile 2 (%)	Quartile 3 (%)
6 March	898	11.44	16.83	2	5	13
9 March	1061	10.70	15.57	2	4	13
9 April	1358	11.12	15.09	2	6	15
20 April	1033	10.43	15.21	2	4	12

CPU utilization. The second metric, ψ is the number of VM migrations initiated during the VM placement adaptation.

3.5.2.2 Workload Data

In order to make a simulation-based evaluation applicable in real world, it is crucial to use workload traces from a real system in experiments [12]. Therefore, the performance of *RTDVMC* and other DVMC algorithms have been measured with real Cloud workload traffic traces representing time varying resource utilization. Real workload data is provided as part of the CoMon project, a monitoring infrastructure for PlanetLab [49]. Data of CPU usage of thousands of VMs has been collected every 5 min, while these VMs had been hosted in PMs spread globally across 500 locations. Both algorithms have been tested with the PlanetLab workload data of four different days: 6 March, 9 March, 9 April, and 20 April featuring different sets of varying resource demand over time. The characteristics of different PlanetLab workload data is articulated in Table 3.3.

For each workload, the associated VMs' release time or workload finishing time have been drawn from monthly VMRT traces of real Cloud, namely, Nectar Cloud. Traces of VMs created in Nectar Cloud over a month along with respective release time of those VMs constitutes the monthly VMRT data. The latest available VMRT data of three different months: November 2013, December 2013, and January 2014 have been used for experiments. To explain more, a single day's PlanetLab workload data is tested with Nectar VMRT data of three different months offering diverse VMRT distributions, so that the impact of heterogeneous workload finishing time or release time can be analyzed. Histogram of different months of Nectar VMRT data has been articulated through Figs. 3.1, 3.2, and 3.3. The number of VMs in Nectar VMRT data of a month is greater than the number of VMs in the PlanetLab workload data of a day. Therefore, a uniformly distributed random variable has been used to select a smaller set of VMs from monthly Nectar data to match the number of VMs of the daily PlanetLab data. Uniformly distributed random variable proffers the smaller set of VMs with similar VMRT distribution present in the monthly Nectar data.

3.6 Simulation Results Analysis

SRTDVMC and *RTDVMC* have been simulated under different workload scenarios. Four different days of PlanetLab workload data has been randomly selected (i.e., 6 March, 9 March, 9 April, and 20 April). PlanetLab workload data of every single day featuring varying resource demand over time has then been blended with three diverse sets of VMRT data originated from three different months of Nectar Cloud Data (i.e., Nectar Nov, Nectar Dec, and Nectar Jan) featuring heterogeneous VMRT. Thus, from a single set of daily PlanetLab workload data, three diverse sets of workload data are produced featuring time variant resource demand and diverse workload finishing time, which matches with real Cloud. Both algorithms are reiterated over multiple times for each set of time variant workload representing a unique combination of PlanetLab and Nectar Cloud data, to produce corresponding $\overline{E}_{\text{CDC}}$ and $\overline{\psi}$.

3.6.1 Energy Consumption

Values of $\overline{E}^{R}$ and $\overline{E}^{S}$ representing mean CDC energy consumption by *RTDVMC* and *SRTDVMC*, respectively are highlighted in Fig. 3.7. Let $X^{\overline{E}}$ (3.12) denote the set representing difference between mean energy consumption by *RTDVMC* and mean energy consumption by *SRTDVMC* for different workload scenarios. In other words, $X^{\overline{E}}_{NT,PL}$ represents the minimization of mean energy consumption proffered by *SRTDVMC* compared to *RTDMC* for diverse workloads, as articulated in Table 3.4.

$$X^{\overline{E}} = \left\{ X^{\overline{E}}_{NT,PL} \right\}_{|Nectar|\cdot|PLab|} = \left\{ \overline{E}^{R}_{NT,PL} - \overline{E}^{S}_{NT,PL} \right\}_{|Nectar|\cdot|PLab|} \tag{3.12}$$

From experimental results, as portrayed in Fig. 3.7 and Table 3.4, we can observe that *SRTDVMC* significantly reduces CDC energy consumption compared to existing DVMC algorithm. However, one might reject the superiority of *SRTDVMC* over existing DVMC algorithm based on the argument that no proof of statistical significance has been provided. To address such arguments, in the following section, we have presented diverse statistical testing.

3.6.1.1 Normality Testing

Parametric tests are reported as more powerful than non-parametric tests. Assumption of parametric tests is that data samples are normally distributed. Therefore, prior parametric tests, normality testing is executed. The capability to accurately figure out if a data sample has come from a non-normal distribution, referred to as

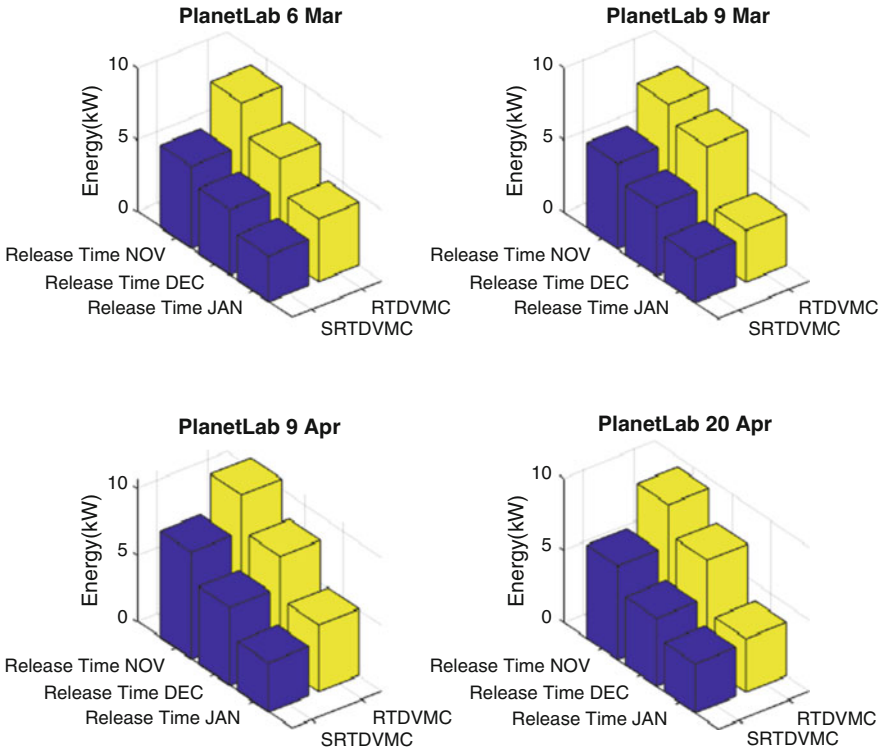

Fig. 3.7 Mean energy consumption of *SRTDVMC* vs *RTDVMC*

Table 3.4 Minimization of mean CDC energy consumption (kW) by *SRTDVMC* compared to *RTDVMC*

		PlanetLab (CPU utilization distribution)			
		6 March	9 March	9 April	20 April
Nectar VMRT	Nov	3.01	2.71	2.75	2.85
	Dec	2.09	2.77	2.3	2.59
	Jan	1.16	0.52	1.3	0.32

power, is the most widespread measure of the strength of a normality test [50]. Chi-square test for normality is not as powerful and unsuitable for small data samples. Kolmogorov-Smirnov (K-S) test is reported to have low power to test normality [51] and has high sensitivity issue with extreme values, which is handled by Lilliefors correction [52]. The S-W normality test is regarded as more powerful than the K-S test even after the Lilliefors correction [53] and recommend as the best option for testing the normality of data [50].

Test statistics for the S-W normality test with $\overline{E}^R$ and $\overline{E}^S$, referred to as $\mathrm{SW}_{\overline{E}}^R$ and $\mathrm{SW}_{\overline{E}}^S$, respectively can be calculated through (3.13)–(3.16) [54, 55]. a_{np} weights are

Table 3.5 Mean CDC energy consumption for *RTDVMC*

np	1	2	3	4	5	6	7	8	9	10	11	12
$\overline{E}^{R}_{(np)}$	3.58	3.72	4.37	5	6.68	7.3	7.53	8.14	8.62	8.75	9.31	10.76

Table 3.6 Mean CDC energy consumption for *SRTDVMC*

np	1	2	3	4	5	6	7	8	9	10	11	12
$\overline{E}^{S}_{(np)}$	3.06	3.21	3.4	3.7	4.59	4.71	4.76	5.74	5.84	5.92	6.46	8.01

available in Shapiro-Wilk Table [56]. Different $\overline{E}^{R}_{(np)}$ and $\overline{E}^{S}_{(np)}$ values are presented in Tables 3.5 and 3.6.

$$\mathrm{SW}^{R}_{\overline{E}} = \left(\sum_{np=1}^{|np|} a_{np} \cdot \left(\overline{E}^{R}_{(|np|+1-np)} - \overline{E}^{R}_{(np)}\right)\right)^2 \Big/ \sum_{w=1}^{|w|}\left(\overline{E}^{R}_{(np)} - \overline{E}^{R}\right)^2 \tag{3.13}$$

$$\mathrm{SW}^{S}_{\overline{E}} = \left(\sum_{np=1}^{|np|} a_{np} \cdot \left(\overline{E}^{S}_{(|np|+1-np)} - \overline{E}^{S}_{(np)}\right)\right)^2 \Big/ \sum_{w=1}^{|w|}\left(\overline{E}^{S}_{(np)} - \overline{E}^{S}\right)^2 \tag{3.14}$$

$$\overline{E}^{R} = \sum_{np=1}^{|np|} \overline{E}^{R}_{(np)} / (|np|) \tag{3.15}$$

$$\overline{E}^{S} = \sum_{np=1}^{|np|} \overline{E}^{S}_{(np)} / (|np|) \tag{3.16}$$

For the S-W normality tests with data samples of $\overline{E}^{R}$ and $\overline{E}^{S}$, the Null Hypothesis is that elements of $\overline{E}^{R}$ and $\overline{E}^{S}$ are normally distributed. We have utilized the software collected from [57] to perform the S-W normality test. For distribution of $\overline{E}^{R}$ and $\overline{E}^{S}$, corresponding p values are found as 0.54 and 0.65 respectively, which are greater than critical value, α as 0.05. Hence, no strong evidence could be found to reject the Null Hypothesis that elements of $\overline{E}^{R}$ and $\overline{E}^{S}$ have come from normal distribution.

3.6.1.2 Parametric Hypothesis Testing and Test Error

Results for normality testing through the S-W normality test have suggested that elements of $\overline{E}^{R}$ and $\overline{E}^{S}$ follow normal distribution, which meets the prior condition of parametric tests. Positive numeric value of every element of $X^{\overline{E}}$ (3.12) articulated in Table 3.4 refers to the fact that energy consumption by *SRTDVMC* is numerically lower compared to *RTDVMC* for different workload scenarios as

presented in experiments. We hence aim to perform parametric hypothesis test to find whether simulation output samples, $X^{\overline{E}}_{NT,PL}$ (3.12) featuring difference between two corresponding means of *RTDVMC* and *SRTDVMC* associated to a unique combination of Nectar and PlanetLab workload is statistically significant. Our sample size is less than 30 and means of no more than two DVMC algorithms (i.e., *SRTDVMC* and *RTDVMC*) would be compared. Therefore, among different parametric tests we opt to use the *t-test*, instead of *Z-test*, *F-test,* and *ANOVA*. Based on the data samples, the *t-tests* can be classified into three groups: One sample, Two Independent Samples and Paired Samples *t-test*. For a specific combination of *NL* and *PL*, corresponding $\overline{E}^{R}_{NT,PL}$ and $\overline{E}^{S}_{NT,PL}$ has a relationship, as $\overline{E}^{R}_{NT,PL}$ and $\overline{E}^{S}_{NT,PL}$ represent $\overline{E}_{\text{CDC}}$ for *RTDVMC* and *SRTDVMC* respectively, under a particular workload distribution scenario. Therefore, the paired two tail *t-test* is performed.

The null hypothesis with the *t-test* is that mean CDC energy consumption with *RTDVMC*, $\overline{E}^{R}$ and mean CDC energy consumption with *SRTDVMC*, $\overline{E}^{S}$ are same, while the alternative hypothesis is that $\overline{E}^{S} < \overline{E}^{R}$. Utilizing (3.17)–(3.19), the test statistic for the *t-test*, denoted by $t_{\overline{X}^{\overline{E}}}$ is found as 2.13 and the corresponding p value is found as 7.10693×10^{-6}, which is lower than critical value, α as 0.05. For clear understanding of the interpretation of the *t-test* result, we have first explained the performed the *t-test* in more details in the following.

$$t_{\overline{X}^{\overline{E}}} = \left(\left(\overline{X}^{\overline{E}} - 0 \right) / \left(\hat{\sigma}_{\overline{X}^{\overline{E}}} \right) \right) \tag{3.17}$$

$$\overline{X}^{\overline{E}} = \left(\left(\sum\nolimits_{np=1}^{|Nectar| \cdot |PLab|} \left(X^{\overline{E}}_{np} \right) \right) / \left(|Nectar| \cdot |PLab| \right) \right) \tag{3.18}$$

$$\hat{\sigma}_{\overline{X}^{\overline{E}}} = \sqrt{\frac{\left(\sum \left(X^{\overline{E}}_{(np)} - \overline{X}^{\overline{E}} \right)^2 / \left((|Nectar| \cdot |PLab|) - 1 \right) \right)}{(|Nectar| \cdot |PLab|)}} \tag{3.19}$$

Previously through S-W normality test results, we have shown that distributions of $\overline{E}^{R}$ and $\overline{E}^{S}$ are normal distributions. It is critical to note that, if we subtract two corresponding elements of two different normal distributions, then the resulting distribution is a normal distribution. Hence, elements of $X^{\overline{E}}$ (3.12) featuring difference between two corresponding means of *RTDVMC* and *SRTDVMC* are normally distributed. Since elements of $X^{\overline{E}}$ are normally distributed, therefore, the distribution of their means, denoted by $\overline{X}^{\overline{E}}$ (3.18), is also normally distributed. Now, assuming the null hypothesis true that $\overline{E}^{R}$ and $\overline{E}^{S}$ are same, the mean of the

distribution of $\overline{X}^{\overline{E}}$ is 0. Nonetheless, utilizing (3.18) with our experimental results articulated in Table 3.4, value of $\overline{X}^{\overline{E}}$ has been found as 2.03 which mismatches with the value of the mean as 0 assuming the null hypothesis is true. As a result, we have then estimated the probability of $\overline{X}^{\overline{E}}$ to be 2.03, given the null hypothesis is true. In order to determine such probability, utilizing (3.17), we have calculated that how many standard deviations far away is our experimental mean (i.e., 2.03) from the distribution mean (i.e., 0), aka the test statistic for the *t-test*.

The test statistic for the *t-test*, denoted by $t_{\overline{X}^{\overline{E}}}$ is found as 2.13 and the corresponding probability is found as 7.10693×10^{-6}. Now, 7.10693×10^{-6} is less than 0.05. In other words, the outcome of the *t-test* shows that if the null hypothesis is true that mean of distribution of $\overline{X}^{\overline{E}}$ is 0, there remains less than 5% chance for $\overline{X}^{\overline{E}}$ to be 2.03. According to the rule of the *t-test*, we can then argue that despite such low probability, since we still have received $\overline{X}^{\overline{E}}$ as 2.03, therefore, the null hypothesis itself cannot be true. So, we retain the alternative hypothesis as true. If $\overline{X}^{\overline{E}}$ (i.e., the mean of minimization of mean CDC energy consumption by *SRTDVMC* compared to *RTDVMC*) was not significant from the perspective of such inferential statistics as the *t-test*, then respective p value would not have been found as lower than 0.05, which gives strong evidence to reject the null hypothesis itself. Hence, we have provided evidence through utilizing inferential statistics that the performance improvement through *SRTDVMC* compared to *RTDVMC* in terms of mean CDC energy consumption is statistically significant.

Errors related to the *t-test* can be classified into two groups: Type I error and Type II error. Type I error refers to the total probability of falsely rejecting the null hypothesis while it was true, and Type II error refers to the total probability of falsely rejecting alternative hypothesis while it was true. The null hypothesis has been rejected based on the corresponding probability value of 7.10693×10^{-6}. Hence, the probability is 7.10693×10^{-6} that we have rejected the null hypothesis while it was true, aka Type I error. In the following section, the simulation results in relation to VM migration has been presented.

3.6.2 VM Migration

VM consolidation is applied to regulate CDC energy consumption. However, one major downside of VM consolidation is that VM consolidation is impossible without VM migration, while, increased VM migration increases network overhead. *SRTDVMC* being a multi-objective DVMC algorithm is designed to minimize CDC energy consumption without incurring increased VM migration. In Fig. 3.8, we have illustrated mean total number of VM migrations with *RTDVMC* and *SRTDVMC*, denoted by $\overline{\psi}^{R}$ and $\overline{\psi}^{S}$, respectively. Let $X^{\overline{\psi}}$ (3.20) denotes the set representing difference of mean total number of VM migrations between *RTDVMC* and *SRTDVMC* under different workload scenario, as articulated in Table 3.7.

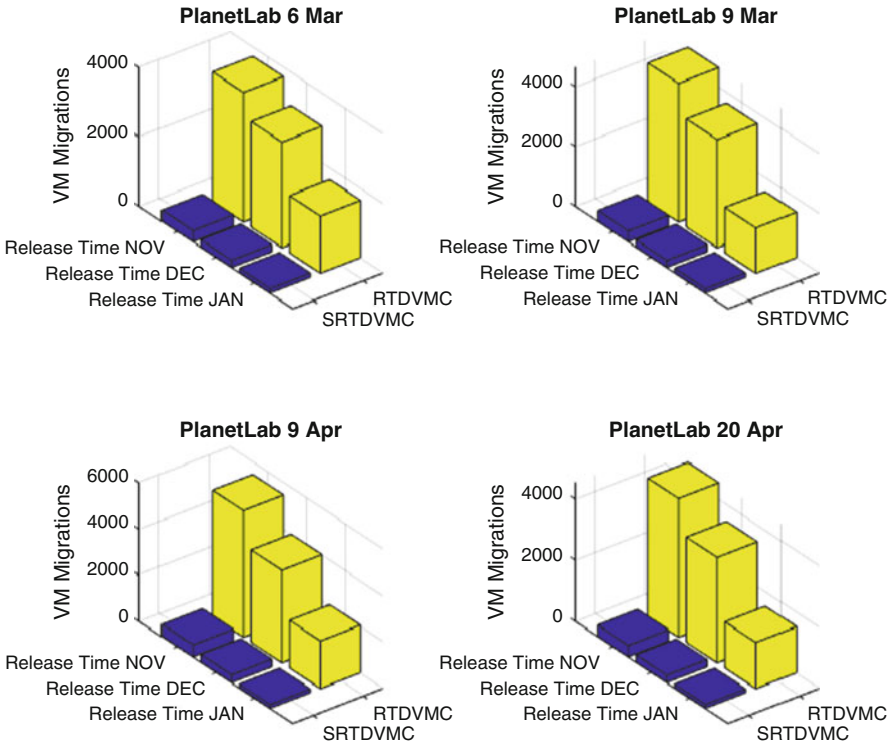

Fig. 3.8 Mean total number of VM migration of *SRTDVMC* vs *RTDVMC*

Table 3.7 Minimization of mean number of VM migration by *SRTDVMC* compared to *RTDVMC*

		PlanetLab (CPU utilization distribution)			
		6 March	9 March	9 April	20 April
Nectar VMRT	Nov	3416	4279	4951	4162
	Dec	2838	3385	3667	3206
	Jan	1531	1381	1908	1390

$$X^{\overline{\psi}} = \left\{ X^{\overline{\psi}}_{NT,PL} \right\}_{|Nectar| \cdot |PLab|} = \left\{ \overline{\psi}^{R}_{NT,PL} - \overline{\psi}^{S}_{NT,PL} \right\}_{|Nectar| \cdot |PLab|} \tag{3.20}$$

From Table 3.7, we can observe that *SRTDVMC* outperforms *RTDVMC* in terms of mean of total number of VM migration. One might argue that such improvement is merely a random event and the results are not statistically significant. To rebut such argument, we have performed the *t-test* on experimental results to check if results are statistically significant or not. One critical point to note that the *t-*

Table 3.8 Mean of total number of VM migration for *RTDVMC*

np	1	2	3	4	5	6	7	8	9	10	11	12
$\overline{\psi}^{R}_{(np)}$	1511	1524	1654	2066	3046	3444	3619	3738	3999	4535	4639	5472

Table 3.9 Mean of total number of VM migration for *SRTDVMC*

np	1	2	3	4	5	6	7	8	9	10	11	12
$\overline{\psi}^{S}_{(np)}$	120	123	143	157	208	234	238	322	332	360	373	521

test cannot prove statistical significance of target data, if data is not normally distributed. To address that issue, normality testing is required to prove that the data representing mean minimization of CDC energy consumption obtained through *SRTDVMC* compared to existing literature as portrayed in Fig. 3.8 is normally distributed. In the following section, we have discussed normality testing performed on our experimental results.

3.6.2.1 Normality Testing

We have applied the S-W normality test with set of mean of total number of VM migration by *RTDVMC* and *SRTDVMC*, denoted by $\overline{\psi}^{R}$ and $\overline{\psi}^{S}$, respectively. Test statistics for the S-W normality test with $\overline{\psi}^{R}$ and $\overline{\psi}^{S}$, referred to as $SW^{R}_{\overline{\psi}}$ and $SW^{S}_{\overline{\psi}}$, respectively can be calculated through (3.21)–(3.24) [54, 55]. a_{np} weights are available in Shapiro-Wilk Table [56]. Different $\overline{\psi}^{R}_{(np)}$ and $\overline{\psi}^{S}_{(np)}$ values are presented in Tables 3.8 and 3.9.

$$SW^{R}_{\overline{\psi}} = \left(\sum_{np=1}^{|np|} a_{np} \cdot \left(\overline{\psi}^{R}_{(|np|+1-np)} - \overline{\psi}^{R}_{(np)}\right)\right)^2 \Big/ \sum_{np=1}^{|np|} \left(\overline{\psi}^{R}_{(np)} - \overline{\overline{\psi}}^{R}\right)^2 \tag{3.21}$$

$$SW^{S}_{\overline{\psi}} = \left(\sum_{np=1}^{|np|} a_{np} \cdot \left(\overline{\psi}^{S}_{(|np|+1-np)} - \overline{\psi}^{S}_{(np)}\right)\right)^2 \Big/ \sum_{np=1}^{|np|} \left(\overline{\psi}^{S}_{(np)} - \overline{\overline{\psi}}^{S}\right)^2 \tag{3.22}$$

$$\overline{\overline{\psi}}^{R} = \sum_{np=1}^{|np|} \overline{\psi}^{R}_{(np)} / (|np|) \tag{3.23}$$

$$\overline{\overline{\psi}}^{S} = \sum_{np=1}^{|np|} \overline{\psi}^{S}_{(np)} / (|np|) \tag{3.24}$$

The null hypothesis for the S-W normality tests with data samples of $\overline{\psi}^R$ and $\overline{\psi}^S$ is that data is normally distributed. For distribution of $\overline{\psi}^R$ and $\overline{\psi}^S$, corresponding p values are found as 0.42 and 0.37 respectively, which are greater than critical value, α as 0.05. Hence, no strong evidence could be found to reject Null Hypothesis that elements of $\overline{\psi}^R$ and $\overline{\psi}^S$ have come from normal distribution. As such, in the following section we have proceeded with the *t-test* to verify if the reduction of VM migration by *SRTDVMC* compared to *RTDVMC* as obtained through experiments is statistically significant.

3.6.2.2 Parametric Hypothesis Testing and Test Error

Previously, we have explained the reason of choosing the two sample paired *t-test*. We aim to prove that minimization of mean of total number of VM migration by *SRTDVMC* compared to *RTDVMC* is statistically significant. As null hypothesis, we assume that the opposite is true. Hence, the null hypothesis is that mean of total number of VM migration, $\overline{\psi}^R$ and mean of total number of VM migration with *SRTDVMC*, $\overline{\psi}^S$ are same, while the alternative hypothesis is that $\overline{\psi}^S < \overline{\psi}^R$. Utilizing (3.25)–(3.27), the test statistic for the *t-test*, denoted by $t_{\overline{X}^{\overline{\psi}}}$ is found as 2.48 and the corresponding p value is found as 1.64×10^{-6}, which is lower than critical value, α as 0.05. In the following, we have elaborately discussed the interpretation of the *t-test* result.

$$t_{\overline{X}^{\overline{\psi}}} = \left(\left(\overline{X}^{\overline{\psi}} - 0\right) / \left(\hat{\sigma}_{\overline{X}^{\overline{\psi}}}\right)\right) \tag{3.25}$$

$$\overline{X}^{\overline{\psi}} = \left(\left(\sum_{np=1}^{|Nectar|\cdot|PLab|} \left(X_{np}^{\overline{\psi}}\right)\right) / \left(|Nectar| \cdot |PLab|\right)\right) \tag{3.26}$$

$$\hat{\sigma}_{\overline{X}^{\overline{\psi}}} = \sqrt{\frac{\left(\sum \left(X_{(np)}^{\overline{\psi}} - \overline{X}^{\overline{\psi}}\right)^2 / \left(\left(|Nectar| \cdot |PLab|\right) - 1\right)\right)}{\left(|Nectar| \cdot |PLab|\right)}} \tag{3.27}$$

In the earlier section, we have shown that such distributions as $\overline{\psi}^R$ and $\overline{\psi}^S$ are normal distributions. As a result, elements of $X^{\overline{\psi}}$ (3.20) are normally distributed. Furthermore, since elements of $X^{\overline{\psi}}$ are normally distributed, therefore, distribution of their means, denoted by $\overline{X}^{\overline{\psi}}$ (3.26) is also normally distributed. Now, under the given null hypothesis that $\overline{\psi}^R$ and $\overline{\psi}^S$ are same, the mean of the distribution $\overline{X}^{\overline{\psi}}$ is 0. Applying the results articulated in Table 3.7, into (3.26), $\overline{X}^{\overline{\psi}}$ is found as 3010. We have then determined the probability of $\overline{X}^{\overline{\psi}}$ to be 3010, under the scenario that null

hypothesis is true (i.e., $\overline{X}^{\overline{\psi}}$ is 0). To determine that probability, we have estimated the distance between our experimental mean (i.e., 3010) and the distribution mean (i.e., 0) in terms of standard deviation, aka the test statistic for the *t-test*, $t_{\overline{X}^{\overline{\psi}}}$ (3.25).

$t_{\overline{X}^{\overline{\psi}}}$ is found as 2.48 and the corresponding probability is found as 1.64×10^{-6}. Now, 1.64×10^{-6} is less than 0.05. In other words, the outcome of the *t-test* shows that given the null hypothesis is true, there is less than 5% chance of $\overline{X}^{\overline{\psi}}$ to be 3010. Nevertheless, despite such low probability, since we still have received $\overline{X}^{\overline{\psi}}$ as 3010, therefore, we can rebut that the null hypothesis itself is not true. Hence, we retain the alternative hypothesis as true. If the mean of minimization of mean CDC energy consumption by *SRTDVMC* compared to *RTDVMC* was insignificant, then respective p value would not have been found as lower than 0.05. Thus, through the *t-test* results we have provided evidence that the reduction of VM migration by *SRTDVMC* compared to *RTDVMC* is statically significant. Since, the null hypothesis has been rejected based on the estimated probability of 1.64×10^{-6}, therefore, the probability of false rejection of null hypothesis while it was true, aka Type I error is 1.64×10^{-6}, which is very low. Experimental results also reveal several critical aspects as discussed in the following section.

3.7 Observations

Observation 1: From experiments results portrayed in Figs. 3.7 and 3.8, we can observe that such traditional DVMC algorithm as *RTDVMC* lacks in performance compared to *SRTDVMC* with the presence of state-of-the-art highly energy proportional PMs. Performance lacking by traditional DVMC algorithm is attributed to its flawed working principal that maximum energy efficiency is attainable through maximum load on PMs.

Observation 2: DVMC algorithm reduces energy consumption through VM migration, which detrimentally affects QoS. As such, concomitant minimization of energy consumption and VM migration are confronting objectives. Hence, developing a DVMC algorithm, which optimizes energy efficiency without increasing the number of VM migration is a much greater challenge than designing an algorithm that merely focuses on the former aspect and ignores the later aspect. *SRTDVMC* being designed to optimize both aspects, only migrates a VM if the respective benefit is greater than the corresponding cost. Our research outcome, as illustrated in Figs. 3.7 and 3.8, substantiates the success of such strategic VM consolidation technique of *SRTDVMC* in both aspects.

Observation 3: Experimental results reveal that energy consumption and VM migration correspond to VMRT. The number of PMs has not been altered. PlanetLab workload data of every single day has been combined with three different VMRT distributions of Nectar Cloud. Hence, from one single set of PlanetLab data of a day, three different sets of workload data have been created representing same number of VMs and same varying resource demand, but different VMRT distributions and performances of both algorithms change with the change in VMRT distributions. The underlying reason is that considering the entire lifetime, aggregated resource consumption by a VM with a relatively large VMRT is likely to be greater than the aggregated resource consumption by a VM with smaller VMRT. In addition, longer VMRT provides more time resulting into more likelihood of VM migrations.

Observation 4: Our research has highlighted that a correlation exists between energy consumption and VMRT. Existence of such correlation has also been found as true between number of VM migrations and VMRT. To elucidate further, one common pattern with both *SRTDVMC* and *RTDVMC* is unfolded that for any day's PlanetLab data, VMRT distribution of November 2013 displays the highest energy consumption and highest number of VM migration, while VMRT distribution of January 2014 proffers to the lowest energy consumption and lowest number of VM migration (Figs. 3.7 and 3.8). The answer lies within VMRT distributions of these months (Figs. 3.1, 3.2, and 3.3). Considering the sum of VMRT of all VMs, VMRT distribution of November 2013 consists of total 8343 days, which is the highest among all months, while VMRT distribution of January 2014 consists of total 727 days, which is the lowest among all months. Since, November 2013 represents the highest total workload duration resulting into highest total resource utilization; therefore, maximum energy consumption is observed for November 2013. Due to maximum duration of total workload existence, number of VM migrations is also found as maximum with November 2013. Similarly, since January 2014 features the lowest total workload duration resulting into lowest total resource utilization, hence, minimum energy consumption is observed for January 2014. Minimum duration of workload existence results into lowest number of VM migration for January 2014.

Observation 5: Experimental results in Figs. 3.7 and 3.8 also present the fact that energy consumption and VM migration are affected by the change in resource demand. Nectar VMRT data of one single month is blended with four different days of PlanetLab data resulting into four different sets of workloads, representing same VMRT, but different sets of resource demand and the performance of both algorithms change due to the variation in resource demand. The reason can be explained through Eqs. (3.8) and (3.9), showing that energy consumption is a function of CPU utilization, while the utilization refers to the sum the resource demand (3.3). Hence, energy consumption is affected by the change in resource demand. The reason of observing the change in total VM migrations with the change in resource demand is that further opportunities of VM consolidation arises as the resource demand changes. DVMC algorithm keeps capitalizing such consolidation

opportunities through further VM migration and hence, number of VM migration changes with the change in resource demand.

Observation 6: Furthermore, energy consumption and VM migration are associated with number of VMs. PlanetLab 9 April features the highest number of VMs, while PlanetLab 6 March holds the lowest number of VMs, which reflects on energy consumption (Fig. 3.7) and VM migration (Fig. 3.8). Higher the number of VMs, higher is the energy consumption and VM migration. The reason is that more VMs consume more resources, resulting into higher energy consumption and more VMs would normally contribute to a greater number of VM migrations. In the following section, we have summarized our research.

3.8 Conclusions and Future Work

3.8.1 Conclusions

While correlation exists between VMRT and energy consumption, traditional DVMC algorithms except *RTDVMC* do not consider heterogenous VMRT in VM consolidation decision process. Furthermore, existing algorithms consolidate VMs in as few PMs as possible based on the premise that optimal energy efficiency can be achieved with maximum load on PM. However, for state-of-the-art PMs, energy efficiency rather drops above 70% load level. Combining lack of consideration of heterogeneous VMRT and ignoring changed energy-efficient characteristics of underlying PMs, existing DVMC algorithms lack in performance in the context of real Cloud scenario with heterogeneous VMRT and state-of-the-art PMs.

RTDVMC considers heterogeneous VMRT. However, issues with *RTDVMC* are twofold – first, it does not take the changed energy-efficiency characteristics of modern PMs into account and second, it only aims to minimize energy consumption without considering VM migration minimization. VM migration, nonetheless, increases network overload causing degraded QoS and increased energy consumption by networking equipment. VM migration being an unavoidable part of VM consolidation, minimizing both energy consumption and VM migrations at the same time are confronting objectives. As such, in this paper, we have brought forth a novel multi-objective DVMC algorithm, namely, *SRTDVMC*, which aims to reduce VM migrations without compromising energy efficiency. Consideration of heterogeneous VMRT values in VM consolidation decision process enables *SRTDVMC* to be more energy efficient. On top of that, contrast to *RTDVMC*, *SRTDVMC* incorporates consideration both benefit and cost prior any VM migration. As a result, it is robust against the changed energy-efficiency characteristics of underlying PMs and can reduce VM migration without compromising energy-efficiency compared to *RTDVMC*.

Performance of *SRTDVMC* has been tested through the most popular Cloud-based simulation tool, namely, CloudSim, in the context of hundreds of different cutting-edge PMs and thousands of VMs representing heterogenous VMRT of real Nectar Cloud, as the assigned workload reflects real Cloud workload obtained from PlanetLab. The empirical outcome exhibits the superiority of *SRTDVMC* over *RTDVMC* in both metrics – CDC energy consumption and VM migration. Three key elements are extracted from our research. First, based on our experiments, VMRT impacts on both aspects – energy consumption and VM migration, and hence, DVMC algorithms are needed to be developed considering the presence of heterogeneous VMRT. Second, such working principal of existing algorithms that maximum energy efficiency is achievable at maximum load on PM is found as false for state-of-the-art PMs, resulting into performance inefficiencies. Our proposed *SRTDVMC* algorithm addresses this issue. Third, simulation results show that if corresponding cost and benefit are considered prior VM migration, then concomitant optimization of both aspects – reduction of energy consumption and VM migration – can be achieved. In the following section, we have suggested several future research pathways to further improve the energy-efficient management of CDC.

3.8.2 Future Work

Accurate estimation of VMRT information plays an important role in minimizing energy consumption through release time-based DVMC algorithm. To explain further, if input VMRT value (i.e., VMRT value given as input in the system) is greater/lower than the true VMRT value (i.e., the authentic time when the VM would truly be removed from CDC), then such decisions as source PM selection, migrating VMs selection and destination PM selection taken on the basis of VMRT value would also be inaccurate, resulting into inefficient performance. Diverse research pathways can be examined in this regard. Advanced machine learning based techniques, neural networks and so forth can be embodied in automated release time-based DVMC algorithms to generate *PVMRT*. Each technique would consist its own set of advantages and disadvantages. It would be an interesting research problem to measure the change in system performance with the change in accuracy of input VMRT.

Different categories of DVMC algorithms can be found from literature, each comes with its own set of advantages and disadvantages. Scope of such diverse heuristic and meta-heuristic DVMC algorithms is yet to be explored for heterogeneous workload coupled with heterogeneous VMRT and state-of-the-art highly energy proportional PMs. As part of future work, we aim to explore that research domain.

References

1. M. Smolaks, *Google Data Center Chief: Go Green Because Your Customers Care* (Data Centre Dynamics Ltd., London, 2014) Available from: http://www.datacenterdynamics.com/content-tracks/design-build/google-data-center-chief-go-green-because-your-customers-care/91942.fullarticle
2. DatacenterDynamics: 3M: The future of data centers will depend on cooling technologies [A detailed look at immersion cooling, with a little help from DCD] (2018). Available from: http://www.datacenterdynamics.com/content-tracks/power-cooling/3m-the-future-of-data-centers-will-depend-on-cooling-technologies/99977.article
3. A. Shehabi, S.J. Smith, N. Horner, I. Azevedo, R. Brown, J. Koomey, et al., *United States Data Center Energy Usage Report* (Lawrence Berkley National Laboratory, Berkley, 2016)
4. M. Avgerinou, P. Bertoldi, L. Castellazzi, Trends in data centre energy consumption under the European code of conduct for data centre energy efficiency. Energies **10**(10), 1470 (2017)
5. European Commission: Code of conduct for energy efficiency in data centres (2016). Available from: https://ec.europa.eu/jrc/en/energy-efficiency/code-conduct/datacentres
6. A. Mark, B. Paolo, B. John, N. Liam, R. Andre, T. Robert, *2018 Best Practice Guidelines for the EU Code of Conduct on Data Centre Energy Efficiency* (European Union, 2018)
7. NECTAR CLOUD 2018. Available from: https://nectar.org.au/research-cloud/
8. F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, et al., Using ant colony system to consolidate VMs for green cloud computing. IEEE Trans. Serv. Comput. **8**(2), 187–198 (2015)
9. F. Farahnakian, R. Bahsoon, P. Liljeberg, T. Pahikkala, (eds.), Self-adaptive resource management system in IaaS clouds, in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD), June 27–July 2 2016*, (IEEE, 2016)
10. F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N.T. Hieu, H. Tenhunen, Energy-aware VM consolidation in cloud data centers using utilization prediction model. IEEE Trans. Cloud Comput. **7**(2), 524–536 (2019)
11. A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Futur. Gener. Comput. Syst. **28**(5), 755–768 (2012)
12. A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr. Comput. **24**(13), 1397–1420 (2012)
13. M.H. Fathi, L.M. Khanli, Consolidating VMs in green cloud computing using harmony search algorithm, in *Proceedings of the 2018 International Conference on Internet and e-Business*, (ACM, Singapore, 2018), pp. 146–151
14. A. Mosa, R. Sakellariou, (eds.), Virtual machine consolidation for cloud data centers using parameter-based adaptive allocation, in *Proceedings of the Fifth European Conference on the Engineering of Computer-Based Systems*, (ACM, 2017)
15. D. Alsadie, E.J. Alzahrani, N. Sohrabi, Z. Tari, A.Y. Zomaya, (eds.), DTFA: A dynamic threshold-based fuzzy approach for power-efficient VM consolidation, in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), 1–3 Nov 2018*, (IEEE, 2018)
16. K. Ajmera, T.K. Tewari, (eds.), Greening the cloud through power-aware virtual machine allocation, in *2018 Eleventh International Conference on Contemporary Computing (IC3), 2–4 Aug 2018*, (IEEE, 2018)
17. Y. Liu, X. Sun, W. Wei, W. Jing, Enhancing energy-efficient and QoS dynamic virtual machine consolidation method in cloud environment. IEEE Access **6**, 31224–31235 (2018)
18. H. Wang, H. Tianfield, Energy-aware dynamic virtual machine consolidation for cloud datacenters. IEEE Access **6**, 15259–15273 (2018)

19. Y. Chang, C. Gu, F. Luo, (eds.), Energy efficient virtual machine consolidation in cloud datacenters, in *2017 4th International Conference on Systems and Informatics (ICSAI), 11–13 Nov 2017*, (IEEE, 2017)
20. S.B. Shaw, J.P. Kumar, A.K. Singh, (eds.), Energy-performance trade-off through restricted virtual machine consolidation in cloud data center, in *2017 International Conference on Intelligent Computing and Control (I2C2), 23–24 June 2017*, (IEEE, 2017)
21. D. Alsadie, Z. Tari, E.J. Alzahrani, A.Y. Zomaya, (eds.), LIFE: A predictive approach for VM placement in cloud environments, in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), 30 Oct–1 Nov 2017*, (IEEE, 2017)
22. M.A.H. Monil, A.D. Malony, (eds.), QoS-aware virtual machine consolidation in cloud datacenter, in *2017 IEEE International Conference on Cloud Engineering (IC2E), 4–7 Apr 2017*, (IEEE, 2017)
23. E. Arianyan, (ed.), Multi objective consolidation of virtual machines for green computing in Cloud data centers, in *2016 8th International Symposium on Telecommunications (IST), 27–28 Sept 2016*, (IEEE, 2016)
24. R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya, CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. **41**(1), 23–50 (2011)
25. The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, *CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services [Internet]* (Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computer Science and Software Engineering The University of Melbourne, Melbourne, n.d.) Available from: http://cloudbus.org/cloudsim/
26. M.A. Khan, A. Paplinski, A.M. Khan, M. Murshed, R. Buyya, Exploiting user provided information in dynamic consolidation of virtual machines to minimize energy consumption of cloud data centers, in *Third International Conference on Fog and Mobile Edge Computing (FMEC), 23–26 Apr 2018, Barcelona, Spain*, (IEEE, 2018)
27. S.S. Masoumzadeh, H. Hlavacs, A gossip-based dynamic virtual machine consolidation strategy for large-scale cloud data centers, in *Proceedings of the Third International Workshop on Adaptive Resource Management and Scheduling for Cloud Computing, Chicago, IL, USA*, (ACM, 2016), pp. 28–34
28. Standard Performance Evaluation Corporation, *SPECpower_ssj2008* (Standard Performance Evaluation Corporation, 2011) Available from: http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110124-00338.html
29. Standard Performance Evaluation Corporation, *SPECpower_ssj2008* (Standard Performance Evaluation Corporation, 2011) Available from: http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110124-00339.html
30. Standard Performance Evaluation Corporation, *SPECpower_ssj2008* (Standard Performance Evaluation Corporation, 2017) Available from: https://www.spec.org/power_ssj2008/results/res2017q4/power_ssj2008-20171010-00789.html
31. Standard Performance Evaluation Corporation, *SPECpower_ssj2008* (Standard Performance Evaluation Corporation, 2017) Available from: https://www.spec.org/power_ssj2008/results/res2017q4/power_ssj2008-20171010-00790.html
32. Standard Performance Evaluation Corporation, *SPECpower_ssj2008* (Standard Performance Evaluation Corporation, 2017) Available from: https://www.spec.org/power_ssj2008/results/res2017q4/power_ssj2008-20171009-00787.html
33. J. von Kistowski, H. Block, J. Beckett, K.-D. Lange, J.A. Arnold, S. Kounev, (eds.), Analysis of the influences on server power consumption and energy efficiency for CPU-intensive workloads, in *Proceedings of the 6th ACM/SPEC International Conference on Performance Engineering*, (ACM, 2015)
34. J. von Kistowski, J. Beckett, K.-D. Lange, H. Block, J.A. Arnold, S. Kounev, (eds.), Energy efficiency of hierarchical server load distribution strategies, in *2015 IEEE 23rd International Symposium on; Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, (IEEE, 2015)

35. D. Wong, M. Annavaram, (eds.), Implications of high energy proportional servers on cluster-wide energy proportionality, in *2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, (IEEE, 2014)
36. J. Gustedt, E. Jeannot, M. Quinson, Experimental methodologies for large-scale systems: A survey. Parallel Process. Lett. **19**(3), 399–418 (2009)
37. M.A. Khan, A. Paplinski, A.M. Khan, M. Murshed, R. Buyya, *Dynamic Virtual Machine Consolidation Algorithms for Energy-Efficient Cloud Resource Management: A Review*, Sustainable Cloud and Energy Services (Springer, 2018), pp. 135–165
38. H. Shen, L. Chen, CompVM: A complementary VM allocation mechanism for cloud systems. IEEE ACM Trans. Netw. **26**(3), 1348–1361 (2018)
39. D. Wong, (ed.), Peak efficiency aware scheduling for highly energy proportional servers, in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, (IEEE, 2016)
40. R. Nasim, A.J. Kassler, (eds.), A robust Tabu Search heuristic for VM consolidation under demand uncertainty in virtualized datacenters, in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, (IEEE, 2017)
41. F. Ahamed, S. Shahrestani, B. Javadi, (eds.), Security aware and energy-efficient virtual machine consolidation in cloud computing systems, in *2016 IEEE Trustcom/BigDataSE/ISPA, 23–26 Aug 2016*, (IEEE, 2016)
42. D. Deng, K. He, Y. Chen, (eds.), Dynamic virtual machine consolidation for improving energy efficiency in cloud data centers, in *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS), 17–19 Aug 2016*, (IEEE, 2016)
43. G.B. Fioccola, P. Donadio, R. Canonico, G. Ventre, (eds.), Dynamic routing and virtual machine consolidation in green clouds, in *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), 12–15 Dec 2016*, (IEEE, 2016)
44. M. Khelghatdoust, V. Gramoli, D. Sun, (eds.), GLAP: Distributed dynamic workload consolidation through gossip-based learning, in *2016 IEEE International Conference on Cluster Computing (CLUSTER)*, (IEEE, 2016)
45. A. Pahlevan, Y.M. Qureshi, M. Zapater, A. Bartolini, D. Rossi, L. Benini, et al., (eds.), Energy proportionality in near-threshold computing servers and cloud data centers: Consolidating or not? in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, (IEEE, 2018)
46. X. Peng, B. Pernici, M. Vitali, (eds.) *Virtual Machine Profiling for Analyzing Resource Usage of Applications* (Springer International Publishing, Cham, 2018)
47. M.H. Ferdaus, M. Murshed, R.N. Calheiros, R. Buyya, (eds.), Virtual machine consolidation in cloud data centers using ACO metaheuristic, in *European Conference on Parallel Processing*, (Springer, 2014)
48. Amazon Web Services: Amazon EC2 instance types (2017). Available from: https://aws.amazon.com/ec2/instance-types/
49. K. Park, V.S. Pai, CoMon: A mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper. Syst. Rev. **40**(1), 65–74 (2006)
50. A. Ghasemi, S. Zahediasl, Normality tests for statistical analysis: A guide for non-statisticians. Int. J. Endocrinol. Metab. **10**(2), 486–489 (2012)
51. H.C. Thode, *Testing for Normality* (CRC Press, Boca Raton, 2002) 368 p
52. J. Peat, B. Barton, *Medical Statistics: A Guide to Data Analysis and Critical Appraisal* (John Wiley & Sons, 2008)
53. D.J. Steinskog, D.B. Tjøstheim, N.G. Kvamstø, A cautionary note on the use of the Kolmogorov–Smirnov test for normality. Mon. Weather Rev. **135**(3), 1151–1157 (2007)
54. Z. Charles, Real statistics using Excel [Internet]. [cited 2018]. Available from: http://www.real-statistics.com/tests-normality-and-symmetry/statistical-tests-normality-symmetry/shapiro-wilk-test/
55. M.E. Clapham, 9: Shapiro-Wilk test [video on the Internet] (2016) [updated 18 Jan 2016]. Available from: https://www.youtube.com/watch?v=dRAqSsgkCUc
56. Z. Charles, Real statistics using Excel [Internet]. [cited 2018]. Available from: http://www.real-statistics.com/statistics-tables/shapiro-wilk-table/
57. Z. Charles, Real statistics using Excel [Internet]. [cited 2019]. Available from: http://www.real-statistics.com/free-download/real-statistics-software/