



ThermoSim: Deep learning based framework for modeling and simulation of thermal-aware resource management for cloud computing environments

Sukhpal Singh Gill^{a,k,*}, Shreshth Tuli^b, Adel Nadjaran Toosi^c, Felix Cuadrado^{a,d}, Peter Garraghan^e, Rami Bahsoon^f, Hanan Lutfiyya^g, Rizos Sakellariou^h, Omer Ranaⁱ, Schahram Dustdar^j, Rajkumar Buyya^k

^a School of Electronic Engineering and Computer Science, Queen Mary University of London, UK

^b Department of Computer Science and Engineering, Indian Institute of Technology (IIT), Delhi, India

^c Faculty of Information Technology, Monash University, Clayton, Australia

^d Technical University of Madrid (UPM), Spain

^e School of Computing and Communications, Lancaster University, UK

^f School of Computer Science, University of Birmingham, Birmingham, UK

^g Department of Computer Science, University of Western Ontario, London, Canada

^h School of Computer Science, University of Manchester, Oxford Road, Manchester, UK

ⁱ School of Computer Science and Informatics, Cardiff University, Cardiff, UK

^j Distributed Systems Group, Vienna University of Technology, Vienna, Austria

^k Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia

ARTICLE INFO

Article history:

Received 28 January 2020

Revised 2 March 2020

Accepted 6 April 2020

Available online 15 April 2020

Keywords:

Cloud computing
Resource management
Thermal-aware
Simulation
Deep learning
Energy

ABSTRACT

Current cloud computing frameworks host millions of physical servers that utilize cloud computing resources in the form of different virtual machines. Cloud Data Center (CDC) infrastructures require significant amounts of energy to deliver large scale computational services. Moreover, computing nodes generate large volumes of heat, requiring cooling units in turn to eliminate the effect of this heat. Thus, overall energy consumption of the CDC increases tremendously for servers as well as for cooling units. However, current workload allocation policies do not take into account effect on temperature and it is challenging to simulate the thermal behavior of CDCs. There is a need for a thermal-aware framework to simulate and model the behavior of nodes and measure the important performance parameters which can be affected by its temperature. In this paper, we propose a lightweight framework, ThermoSim, for modeling and simulation of thermal-aware resource management for cloud computing environments. This work presents a Recurrent Neural Network based deep learning temperature predictor for CDCs which is utilized by ThermoSim for lightweight resource management in constrained cloud environments. ThermoSim extends the CloudSim toolkit helping to analyze the performance of various key parameters such as energy consumption, service level agreement violation rate, number of virtual machine migrations and temperature during the management of cloud resources for execution of workloads. Further, different energy-aware and thermal-aware resource management techniques are tested using the proposed ThermoSim framework in order to validate it against the existing framework (Thas). The experimental results demonstrate the proposed framework is capable of modeling and simulating the thermal behavior of a CDC and ThermoSim framework is better than Thas in terms of energy consumption, cost, time, memory usage and prediction accuracy.

© 2020 Elsevier Inc. All rights reserved.

* Corresponding author at: School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, UK.

E-mail addresses: s.s.gill@qmul.ac.uk (S.S. Gill), shreshth.cs116@cse.iitd.ac.in (S. Tuli), adel.n.toosi@monash.edu (A.N. Toosi), felix.cuadrado@upm.es (F. Cuadrado), p.garraghan@lancaster.ac.uk (P. Garraghan), r.bahsoon@cs.bham.ac.uk (R. Bahsoon), hanan@csd.uwo.ca (H. Lutfiyya), rizos@manchester.ac.uk (R. Sakellariou), ranaof@cardiff.ac.uk (O. Rana), dustdar@dsg.tuwien.ac.at (S. Dustdar), rbuyya@unimelb.edu.au (R. Buyya).

1. Introduction

Resource management is critical in cloud environment in which resource utilization, power consumption of servers, and storage play important roles. Provisioning and scheduling cloud resources is often based on availability, without considering other crucial

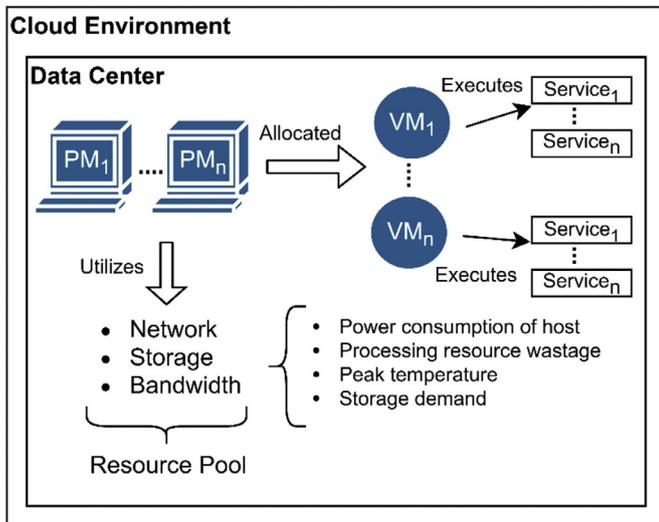


Fig. 1. Interaction of various entities in cloud data center for resource management.

parameters such as resource utilization or the server's thermal characteristics (Chia-Ming et al., 2014). To realize this, a thermal-aware simulator for resource allocation mechanism is required (Tsai and Chen, 2016). The problem of allocating user workloads to a set of Physical Machines (PMs) or Virtual Machines (VMs) and allocating VMs on different server farms adhering to the terms of service as cited in Service Level Agreements (SLAs) and sustaining the Quality of Service (QoS) is stated as the service provisioning issue. Thus, cloud providers focus on developing energy-efficient approaches and policies (Mhedheb et al., 2013).

Thermo-awareness in cloud refers to the consideration of thermal properties, such as thermal temperature of the host, CPU temperature, heat tolerance and thresholds, energy source (i.e. non-renewable vs. renewable), cooling considerations and mechanisms, cost etc. when dynamically managing cloud resources, scheduling and allocating workloads (Qinghui et al., 2008). The explicit consideration of these properties can transform the way the cloud is managed and resources/PMs/VMs are dynamically allocated, leading to more energy-efficient computing and reduced carbon footprint (Lin et al., 2019) (Wu et al., 2019). The consideration of these properties can inform the design and definition of new policies that consider energy and thermal properties as moving targets and calls for dynamic management and optimization of cloud resources (Chaudhry et al., 2015). It is also imperative that the consideration of thermo properties needs to be balanced, traded-off and/or factored along QoS provision; this can be monitored and observed on SLA compliance/violation.

In this work, we consider the thermal characteristics of the host machine focusing on the issue of allocating VMs to hosts in the server farms and assigning workload to the appropriate VMs considering performance parameters (Ilager et al., 2019). The VMs are sorted based on their resource utilization, memory utilization, disk utilization and network utilization as discussed in Section 3.2. The anticipated scheduling policy reduces the energy of the PM, resource utilization with the aid of high-performance distribution strategies (Xiang et al., 2018; Zhou et al., 2020). Fig. 1 illustrates the basic architecture of resource management in cloud computing environments and describes the interaction of various entities in cloud data center for resource management. The entire incoming load of the server farm is distributed among several VMs for execution (Gill et al., 2019). The aggregate workload of the server farm is the finite number of jobs where each job assigned to a few

VMs for execution which in turn are hosted by PMs (Wu and Xu, 2010).

1.1. Motivation and our contributions

A well-known cloud simulator, CloudSim toolkit (Calheiros et al., 2011) is available, which allows to model and simulates cloud computing environments that resemble real-world infrastructure elements. However, CloudSim toolkit does not include thermal aspects of a data center. Therefore, there is a need to develop a simulation platform as a benchmark to incorporate thermal characteristics which establish a relationship between theory and practice for thermal-aware resource management. The required simulation framework can combine both utilization and thermal models to perform VM allocation to reduce heat generation and hence energy required in computing systems (Moore et al., 2007; Qinghui et al., 2008; Lazic et al., 2018; Ranganathan and Sharma, 2005; Chaudhry et al., 2015). This more holistic approach allows it to infer more complex patterns of behavior between resource utilization and heat generation to engender a more efficient approach of resource management which in effect increases the performance of the system (Xiang et al., 2018; Ilager et al., 2019; Gill et al., 2019; Gill et al., 2019). The motivation behind this research work is to propose a framework for the simulation of thermal-aware resource management for cloud computing environment, called **ThermoSim**, which benefits by combining compute utilization and physical heat characteristics over existing systems. We develop a lightweight deep learning-based temperature predictor using Recurrent Neural Network (RNN) which utilizes resource consumption metrics to accurately predict temperature for cloud hosts. We extend the base classes of CloudSim toolkit to incorporate thermal parameters into it. We evaluated the feasibility and performance of ThermoSim framework and compared it to another baseline simulator.

The **main contributions** of this research work are:

1. A novel framework called ThermoSim is proposed for thermal-aware resource management for cloud computing environments by extending CloudSim toolkit.
2. In ThermoSim, thermal-aware and utilization based approaches are proposed for scheduling of resources to optimize energy consumption and temperature simultaneously.
3. A lightweight RNN based deep learning predictor for temperature characteristics of cloud hosts is presented for low overhead resource management in ThermoSim
4. Proposed scheduling approaches equipped with efficient energy and thermal-aware policy for enhanced performance of cloud data centers.
5. Validated ThermoSim framework against Thas (Mhedheb et al., 2013) based on different system parameters such as memory, time, cost, energy and prediction accuracy using datasets from Alibaba Cluster and PlanetLab.
6. ThermoSim analyze the performance of existing energy-aware and thermal-aware scheduling policies based on different QoS parameters such as SLA Violation Rate, energy consumption, number of VM migrations and temperature.
7. We propose key future research directions in the context of ThermoSim framework.

Lightweight Testbed/Simulator: ThermoSim is designed to build an experimental testbed/simulator for conducting practical research in the domain of thermal-aware resource management for cloud computing. Researchers can simulate the thermal behavior of the entire data center using ThermoSim and test or validate their approach for thermal-aware resource management before implementing on real CDC (sensors, servers and GPUs).

1.2. Article organization

The rest of the paper is organized as follows: [Section 2](#) presents the related work. [Section 3](#) presents the ThermoSim framework. [Section 4](#) describes the performance evaluation, validation of ThermoSim framework and experimental results. [Section 5](#) presents conclusions, future research directions and open challenges.

2. Related work

The existing frameworks that allow thermal-aware resource scheduling have significant drawbacks in terms of their ability to extract and predict thermal characteristics in a CDC. We have categorized the related work based on non-SLA and SLA aware resource management techniques for thermal management.

2.1. Non-SLA aware resource management

[Kim et al. \(2019\)](#) proposed a Dynamic Thermal Management (DTM) technique, which exploits external computing resources (idle servers) adaptively as well as internal computing resources (free cores of CPU in the server) available in heterogeneous data centers. DTM technique identifies memory intensiveness and usage of VMs if the temperature of a CPU core in a server exceeds a pre-defined thermal threshold and migrates the VMs among CPU cores in the server to maintain the temperature of the server. DTM technique saves energy and improves the performance while satisfying thermal constraints efficiently. DTM technique uses reactive mechanism but not able to predict the temperature variations proactively. Moreover, the impact of temperature variations on SLA has not been identified. [Fu et al. \(2017\)](#) proposed Temperature-Aware Resource Management (TARM) algorithm, which uses Lyapunov Optimization theory to maintain the server temperature without effecting the system reliability. TARM algorithm decreases energy consumed by Computer Room Air Conditioning (CRAC) and servers while imposing server temperature constraints and QoS. Further, a trade-off between system energy consumption and server temperature has been developed, which reported that there is a need of thermal-aware and utilization-based approach to optimize the performance efficiently. TARM algorithm cannot measure the impact of number of VM migrations on system performance and SLA.

[Fu et al. \(2017\)](#) proposed a Dynamic Control Algorithm (DCA) without breaking the average temperature constraints and designed a Server Temperature-Constrained Energy Minimization (STCEM) problem. Further, DCA develops linear and quadratic control policies to solve STCEM problem using Lyapunov optimization, similar to TARM algorithm ([Fu et al., 2017](#)). Further, trade-off between energy and temperature is designed to compare the performance of linear with quadratic control policy and identifies the impact on power usage on system performance during the execution of workloads. In this approach, the impact of energy consumed by cooling components on overall temperature has not been identified. [Wu et al. \(2016\)](#) proposed VM level temperature prediction in Cloud datacenters and measures the impact of CPU temperature on system performance dynamically with/without calibration compared to empirical data. Experimental results show that dynamic CPU temperature modeling with calibration at run time produces more accurate information. In this study, various important performance parameters such as energy, SLA violation rate and their impact on temperature is not discussed. [Pierson et al. \(2019\)](#) formulated Mixed Integer Linear Programming (MILP) for spatio-temporal thermal-aware scheduling while considering dynamics of heat dissipation and production during scheduling of workloads. Further, performance parameters such as energy and execution time are optimized to improve the performance of CDC. In this research work, the impact of number of VM migrations

on system performance and SLA is not discussed. [Oxley et al. \(2016\)](#) proposed an online resource management technique for thermal and energy constrained heterogeneous cloud environment, which uses offline analysis to predict temperature variations at runtime to improve the performance of the system. Earlier temperature prediction helps to execute the workload within their deadline and specified budget. Further, an automatic load balancer is used to balance the load in case of performance degradation while increasing server temperature. This study failed to identify the impact of temperature change on SLA violation rate and energy consumption.

[Liu et al. \(2018\)](#) proposed a thermal and power-aware model which also considers computing, cooling and task migration energy consumption. But due to their modeling limitations they have not considered I/O energy consumption, network transmission energy consumption. Moreover, they have measured parameters at a very coarse granularity. This work failed to identify the impact of temperature and energy consumption on SLA violation rate and number of VM migrations during execution of workloads. [Akbar et al. \(2019\)](#) proposed a game based thermal aware allocation strategy using Cooperative Game Theory and it decreases the imbalance within the CDC by using the concept of cooperative game theory with a Nash-bargaining to assign resources based on thermal profile. A problem in their approach is that they consider a system to be homogeneous and does not incorporate violation rate to categorize tasks and VMs. For heterogeneous environments, their algorithm can lead to contention and poor load balancing. [Khaleel \(2018\)](#) described a thermal-aware load balancing strategy by calculating the shortest distance to cloud resources deployed at different geographical locations and conserving more bandwidth cost. Further, it has been suggested that running servers at different locations can reduce temperature at particular location and improves the health of server. Proposed strategy improves the resource utilization without considering the other type of utilization such as disk, network and memory. This would not be feasible in virtual containers-based cloud services and mobile service providers.

2.2. SLA aware resource management

[Chen et al. \(2017\)](#) proposed thermal-aware VM migration manager to identify the working condition of server based on resource utilization and temperature and recognizes the impact of CPU overheating (caused by chassis fan damage) on system performance. Further, it is shown that the migration of VMs from overloaded server to underloaded server balances the load, reduces the damages to overloaded servers and improves its health and system performance. Proposed technique decreases the number of VM failures and improves the system ability which further reduces SLA violation rate, but this technique has not been identified the impact of temperature variation on SLA violation rate. [Mhedheb et al. \(2013\)](#) proposed a Thermal aware scheduler (Thas) on the CloudSim toolkit, which implements an interface between the hypervisor and the virtual machine in CDC. They have replaced the CloudSim's VMScheduler class which results in their scheduling behavior to be highly data-dependent of the existing class inputs. However, Thas lacks thermal characteristic studies by which they are not able to find the best host location at the time of VM migration. Moreover, their model is restricted in terms of analyzing the CPU and memory loads and the number of VM migrations which significantly impacts the performance of the system in terms of its latency and other QoS parameters. [Xiang et al. \(2018\)](#) proposed a virtual machine scheduling technique for Cloud Datacenters which reduces total energy consumption while holistically managing the components of CDC such as CPU, memory, storage, network and cooling. This research work proposed a ther-

mal and energy-aware model to analyze the temperature variation and distribution in the CDC during execution of workloads. Further, a holistic resource management approach is proposed to reduce the energy consumption of CDC and maintains the temperature of CDC below than critical temperature. The architecture comprises of three sub-components such as workload manager, scheduling manager and cooling manager. Workload manager manages the workloads submitted by user and process for scheduling based on their requirements. Scheduling manager schedules the resources for execution of workloads while maximizing the performance of CDC and minimizes the consumption of energy. Cooling manager maintains the temperature of CDC and save cooling energy by performing VM placement and dynamic migration in an efficient manner. Proposed model updates computing capacity dynamically to improve cooling efficiency and maintains the CPU temperature less than threshold value while adjusting cooling energy to the lowest level to save energy and reduce SLA violation rate. It uses reactive mechanism to maintain temperature but not able to predict the temperature variations proactively. Moreover, the impact of temperature variations on SLA has not been identified.

Rodero et al. (2012) also provided a strategy to allocate VMs using their temperature characteristics. They propose a reactive technique as an alternative to VM migration and DVFS which reduces the activity of one or more VMs by pinning them to specific Cloud Management Portals (CMPs). They profile applications to decide which VM to pin with which CPU. This in effect has significant overhead in terms of CPU, Memory and Time and not able to exploit thermal data completely as it significantly depends on the OS's default dynamic CPU power management which is not aware of the characteristics of other physical machines in the network. Kumar and Raghunathan (2016) presented a suite of heuristics for energy efficiency consolidation and a hybrid scheduling algorithm to maintain the temperature of the CDC and reduces the energy consumption of different servers within the CDC. They propose StaticPPMMax (performance to power metrics using server's peak power consumption) and compute metrics for VM allocation, but do not consider server process sleep state and transition power consumption which makes their approach weak and not scalable for complex workload models. Ilager et al. (2019) proposed an Energy and Thermal-Aware Scheduling (ETAS) algorithm that dynamically consolidates VMs to minimize the overall energy consumption while proactively preventing hotspots. They have extended a class of the CloudSim toolkit which does not allow them to validate their results on fair grounds. Moreover, their algorithm assumes static cooling environment, which may not be versatile to different cooling settings.

2.3. Critical analysis

Table 1 shows the comparison of ThermoSim with existing frameworks. All the above research works have presented thermal-aware scheduling frameworks in cloud computing without considering the thermal-aware and utilization-based resource management simultaneously in a single framework, but it is very important to study the behavior of both resource management approaches together to optimize the different QoS parameters in a controlled and holistic manner. None of the existing works validated against prediction accuracy, memory and time and only two frameworks (Khaleel, 2018; Ilager et al., 2019) considered time for validation. As per literature, there is a no existing framework which considers all the four QoS parameters (SLA Violation Rate, energy consumption, number of VM migrations and temperature) in a single framework. Due to this, the current thermal-aware resource management become inefficient to respond in these situations. Thermal aware techniques provide benefits in some cases, however fail in some other cases for which energy aware ap-

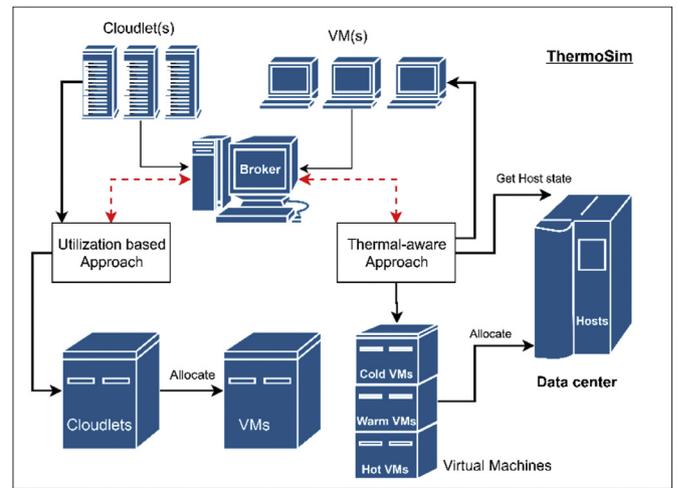


Fig. 2. Architecture of ThermoSim framework.

proaches are required (Gill et al., 2019; Möbius et al., 2014; Li et al., 2018; Wu et al., 2019). Thus, ThermoSim uses an integrated approach leveraging both techniques for optimum results.

Therefore, there is a need to develop a simulation platform as a benchmark: 1) to establish a relationship between theory and practice for thermal-aware resource management, 2) to incorporate thermal characteristics, 3) to schedule resources using thermal-aware and utilization-based resource management simultaneously, 4) to validate the thermal-aware resource management framework against system parameters such as memory, time, cost, energy and prediction accuracy and 5) to test the performance of thermal and energy-aware scheduling policies based on all the four QoS parameters such as SLA Violation Rate, energy consumption, number of VM migrations and temperature. Our proposed ThermoSim framework addresses the challenges of existing frameworks in this research work.

3. ThermoSim framework

This section presents the detailed description of ThermoSim framework. Fig. 2 presents the architecture of the ThermoSim framework, which is based on two different models: 1) Utilization Model 2) Thermal Model. Energy model is an integral part of Utilization model. In the utilization model, cloud workloads are assigned to VMs based on their different types of utilization (resource, network, memory and disk) and energy consumption, while the thermal model considers the thermal characteristics of the host machine and accordingly VMs are scheduled on PMs.

3.1. Energy model

The larger part of the data center energy consumption is contributed by the computing and cooling systems (Singh et al., 2016; Calheiros et al., 2011; Gill et al., 2019; Qinghui et al., 2008; Lazic et al., 2018; Xu and Buyya, 2019; Möbius et al., 2014; Li et al., 2018; Balis et al., 2018; Liu et al., 2012).

$$E_{Total} = E_{Computing} + E_{Cooling} \quad (1)$$

3.1.1. Computing

The computing system consists of hosts and its energy consumption can be defined as follows:

$$E_{Computing} = E_{Processor} + E_{Storage} + E_{Memory} + E_{Network} + E_{Extra} \quad (2)$$

Table 1
Comparison of ThermoSim with exiting frameworks.

Framework	Performance Parameters				Validation					Deep Learning based Temperature Prediction	Resource Management	
	SLA Violation Rate	Energy Consumption	Number of VM Migrations	Temperature	Prediction Accuracy	Memory	Time	Cost	Energy		Thermal-aware Approach	Utilization based Approach
Non-SLA aware Resource Management												
(Liu et al., 2018)		✓		✓					✓		✓	
(Akbar et al., 2019)		✓	✓	✓					✓		✓	
(Khaleel, 2018)		✓	✓				✓				✓	
(Kim et al., 2019)		✓		✓							✓	
(Fu et al., 2017)		✓		✓							✓	
(Fu et al., 2017)		✓		✓							✓	
(Wu et al., 2016)				✓							✓	
(Pierson et al., 2019)		✓					✓				✓	
(Oxley et al., 2016)				✓				✓			✓	
SLA aware Resource Management												
(Mhedheb et al., 2013)	✓	✓		✓					✓		✓	
(Rodero et al., 2012)	✓	✓	✓	✓					✓		✓	
(Kumar and Raghunathan, 2016)	✓	✓	✓						✓		✓	
(Ilager et al., 2019)	✓	✓	✓	✓			✓		✓		✓	
(Chen et al., 2017)	✓	✓	✓	✓					✓		✓	
ThermoSim	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

$E_{Processor}$ represents the processor's energy consumption, which is calculated using [Eq. 3]:

$$E_{Processor} = \sum_{r=1}^{r=cores} (E_{dynamic} + E_{SC} + E_{Leakage} + E_{idle}) \quad (3)$$

where $E_{dynamic}$ represents dynamic energy consumption, E_{SC} represents short-circuit energy consumption, $E_{Leakage}$ represents power loss due to transistor leakage current and E_{idle} represents the energy consumption when processor component is idle. Dynamic energy consumption ($E_{dynamic}$) is calculated using Eq. (4), which is an average of energy consumption calculated using linear and non-linear model.

$$E_{dynamic} = \frac{E_{dynamic}^{linear} + E_{dynamic}^{non-linear}(U_j)}{2} \quad (4)$$

$E_{dynamic}^{linear}$ is the dynamic energy using linear model (Xiang et al., 2018) and it is calculated using Eq. (5):

$$E_{dynamic}^{linear} = CV^2 f \quad (5)$$

where C is capacitance, f is frequency, and V is voltage. $E_{dynamic}^{non-linear}$ is the dynamic energy using non-linear model (Zhao et al., 2018), resource utilization has a non-linear relationship with energy consumption and it is calculated using Eq. (6):

$$E_{dynamic}^{non-linear}(U_j) = \mu_1 \cdot U_j + \mu_2 \cdot U_j^2 \quad (6)$$

Where μ_1 and μ_2 are nonlinear model parameters and U_j is CPU utilization of host h_j .

$$M_{Utilization} = \frac{\text{Total Physical Memory} - (\text{Memory Free} + \text{Memory Buffers} + \text{Cache Memory})}{\text{Total Physical Memory}} \times 100 \quad (13)$$

$E_{Storage}$ represents the energy consumption of storage device, which performs data read and write operations and it is calculated using Eq. (7):

$$E_{Storage} = E_{ReadOperation} + E_{WriteOperation} + E_{idle} \quad (7)$$

where E_{idle} represents the energy consumption when storage component is idle.

E_{Memory} represents the energy consumption of the main memory (RAM/DRAM) and cache memory (SRAM), which is calculated using Eq. (8):

$$E_{Memory} = E_{SRAM} + E_{DRAM} \quad (8)$$

$E_{Network}$ represents the energy consumption of networking equipment such as routers, switches and gateways, LAN cards, which is calculated using Eq. (9):

$$E_{Network} = E_{Router} + E_{Gateway} + E_{LANcard} + E_{Switch} \quad (9)$$

E_{Extra} represents the energy consumption of other parts, including the current conversion loss and others, which is calculated using Eq. (10):

$$E_{Extra} = E_{Motherboard} + \sum_{f=0}^F E_{connector, f} \quad (10)$$

where $E_{Motherboard}$ is energy consumed by motherboard (s) and $\sum_{f=0}^F E_{connector, f}$ is energy consumed by a connector (port) running at the frequency f .

3.1.2. Cooling

The energy model for computing is developed based on energy consumption of different cooling components to maintain the temperature of CDC. $E_{Cooling}$ represents the energy is consumed by cooling devices (compressors, Air Conditioners (AC) and fans) to maintain the temperature of cloud datacenter, which is calculated using Eq. (11):

$$E_{Cooling} = E_{AC} + E_{Compressor} + E_{Fan} \quad (11)$$

3.2. Utilization model

The jobs originate by cloud consumers as demanded services known as "cloud workloads" (Singh and Chana, 2016). These workloads are submitted to the workload queue of the cloud system. The workload is sorted in ascending order of VMs based on resource utilization and placed in the queue to assign workload to virtual machines. *Resource Utilization* is a ratio of an execution time of a workload executed by a particular resource to the total uptime of that resource (Gill et al., 2019). The total uptime of resource is the amount of time available with a cloud resource set for execution of workloads. We have designed the following formula to calculate resource utilization ($R_{Utilization}$) Eq. (12).

$$R_{Utilization} = \sum_{i=1}^n \left(\frac{\text{execution time of a workload executed on } i^{th} \text{ resource}}{\text{total uptime of } i^{th} \text{ resource}} \right) \quad (12)$$

Where n is the number of resources. These VMs are also sorted in terms of their network utilization, memory utilization and disk utilization in descending order i.e. opposite order in which workloads are sorted and placed in the queue. The formula for calculating memory utilization ($M_{Utilization}$) in percentage (Gill et al., 2019) is as follows Eq. (13):

The formula for calculating disk utilization ($D_{Utilization}$) in percentage is as follows Eq. (14):

$$D_{Utilization} = \frac{\text{Total Used}}{\text{Total HD size}} \times 100 \quad (14)$$

$$D_{Utilization} = \frac{\text{Storage Allocation Units} \times \text{Storage Used}}{\text{StorageAllocationUnits} \times \text{Storage Size}} \times 100 \quad (15)$$

The formula for calculating network utilization ($N_{Utilization}$) in percentage (Gill et al., 2019) is as follows Eq. (16):

$$N_{Utilization} = \frac{\text{data bits}}{\text{bandwidth} \times \text{interval}} \times 100 \quad (16)$$

The workloads present in the sub task queues are submitted to the data center broker. Workload classification and assignment are shown in Fig. 3. Thus, cloud workloads are mapped to the VMs based on energy-efficient resource management policy using Cuckoo Optimization based scheduling technique (Gill et al., 2019). Algorithm 1 presents the utilization-based approach which i) sorts the cloud workloads and ii) maps workloads to VMs. Based on the current utilization of VMs, the scheduler allocates the tasks accordingly. To do this, workloads are organized in increasing order of utilization, and VMs in decreasing order with an aim to allocate tasks which lower the utilization i.e. light-weight tasks are executed on VM with high utilization, and vice-versa.

The main idea of Algorithm 1 is described below:

1. Initialize the hosts (PMs) as available at time stamp to 0 i.e. host is available at beginning of scheduling.
2. Sort workloads and VMs based on utilization.
3. Map workloads to VMs.

Algorithm 1 sorts tasks with increasing utilization requirements and sorts VMs with decreasing utilization requirements. This allows high resource requirement tasks to be allocated on

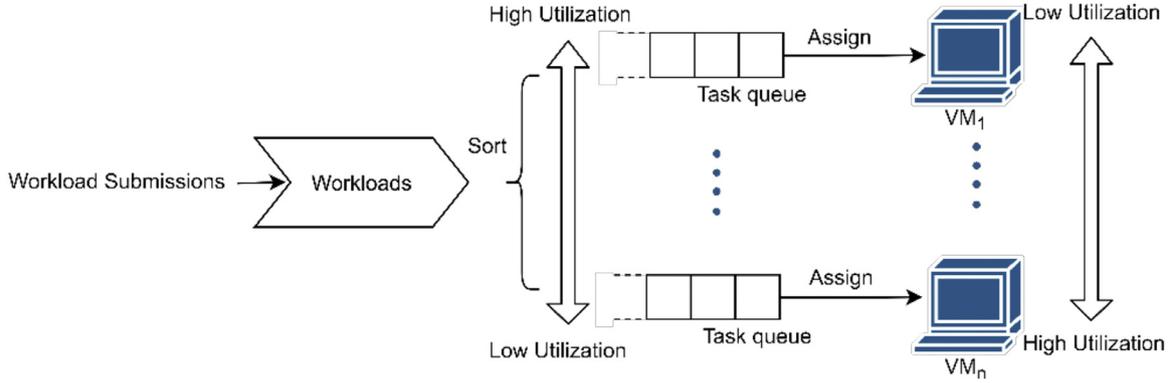


Fig. 3. Workload classification and assignment.

Algorithm 1 Utilization based Approach for Resource Management.

1. **Input:** Number of workloads (T) and number of available resources (V)
2. **Output:** Mapping of each workload to the resource
3. **Function** *UtilizationSort* (*list*, *vm*, *decreasing*)
4. **if** (*vm* == True) **then**
5. Sort list based on increasing order of E_{Total}
6. **return** (*UtilizationSort* (*list*, false))
7. **if** (*decreasing* == True) **then**
8. Sort list based on decreasing order of $R_{Utilization}$
9. **else**
10. Sort list based on increasing order of $R_{Utilization}$
11. Break ties using $M_{Utilization}$
12. Break ties using $D_{Utilization}$
13. Break ties using $N_{Utilization}$
14. **return** (*list*)
15. **Start**
16. Initialize all host list (Number of PMs)
17. Initialize all resource list (Number of VMs)
18. Initialize all workload list (Number of tasks)
19. $T' = UtilizationSort(T, false, false)$ // sort task with increasing utilization (estimating variation in utilization while executing task t)
20. $V' = UtilizationSort(T, true, true)$ // sort VM with decreasing utilization (function to estimate utilization for VM)
21. **for** task t in T' :
22. **for** vm v in V' :
23. **if** t is suitable for v in V' **then**
24. Schedule the task t on VM v
25. **End**

VMs which have low load. This greedy algorithm allows efficient scheduling of tasks to VMs based on their requirements. The sorting algorithm can be implemented using merge sort [$O(n \log n)$], and the allocation of tasks we iterate over all VMs for every task which makes it [$O(n^2)$] in worst case. Here n is maximum of the number of tasks or VMs.

3.3. Thermal model

The idea is to design a scheduling policy for VMs based on the CPUs temperature characteristics. Thus, a thermal model is needed that describes these changes of this parameter when workloads are running on virtual machines. Thermal-aware scheduling considers current temperature and maximum working temperature, that is, the threshold temperature of every machine (Ilager et al., 2019; Moore et al., 2007; Qinghui et al., 2008; Lazic et al., 2018; Ranganathan and Sharma, 2005; Zhou et al., 2017), before making scheduling decisions. Let the maximum threshold temperature of a server machine be T_{over} and let current temperature of a server machine be T_{cu} . T_{over} is the temperature beyond which a machine is overheated. The heuristic chosen for VM scheduling is the difference between threshold and present temperature, as formulated in

Eq. (17). ΔT_{vi} is the temperature variation of the host.

$$\Delta T_{vi} = T_{over} - T_{cu} \quad (17)$$

We used Computer Room Air Conditioning (CRAC) model and RC (where R and C are thermal resistance (k/w) and heat capacity (j/k) of the host respectively) thermal model (Lee et al., 2017; Gill et al., 2019; Moore et al., 2007; Qinghui et al., 2008; Lazic et al., 2018; Ranganathan and Sharma, 2005; Möbius et al., 2014) to design temperature model for calculation of current temperature of CPU (T_{cu}). The following formula is used to calculate the current temperature of CPU Eq. (18).

$$T_{cu} = PR + Temp_{inlet} + T_{initial} \times e^{-RC} \quad (18)$$

where CRAC model is used to calculate inlet temperature ($Temp_{inlet}$) and RC model is used to calculate CPU temperature ($Temp_{CPU}$) and P is the dynamic power of host. $T_{initial}$ is the initial temperature of the CPU. The relationship between a power (energy efficiency) and temperature model (heating model) (Chaudhry et al., 2015) is described in Eq. (18). VMs are separated into various classes as per their temperature attributes, which are then distributed to the hosts based on their temperatures. The VM movement component is directed to guarantee the unwavering quality of the model when a host achieves threshold temperature.

3.4. Deep learning based temperature prediction module

In many CDCs, it is difficult to access the thermal characteristics of the hosts (Zhou et al., 2020). Mostly, the temperature sensors are either too expensive or are unavailable to give accurate temperature information (Saha and Majumdar, 2017; Bakker and Huijsing, 2013). Having noisy temperature data can significantly affect the performance of any thermal-aware simulator including ThermoSim. The thermal model in Section 3.3 requires various metrics like thermal resistance (R), capacitance (C), $T_{initial}$, which may not be available for many CDCs. Moreover, the CRAC and RC models are resource intensive. Thus, there is a requirement of a prediction module that can predict thermal characteristics by observing simpler metrics like memory, CPU, I/O utilization and fan speeds. ThermoSim, thus, can also use a temperature prediction module that uses Recurrent Neural Network (RNN) (Dey and Fathi, 2017) to predict CPU temperatures for different PMs. This is not only useful for CDC where temperature sensors are unavailable or expensive but also when the sensor data is noisy.

To predict the CPU temperature, our deep learning model uses RNN with 4 Gated Recurrent Units (GRU) layers as shown in Fig. 4. The input of the network is a matrix with various features of all PMs. These features include fan speeds, and resource utilization metrics. The output of the network is a vector of temperatures for all PMs. A Recurrent Neural Network (RNN) is a class of artificial neural networks where connections between nodes form a

Table 2
Sample data extracted from Alibaba CDC.

Server ID and Time		Fan (RPM)					Utilization (%)				Temp (°C)
Id	Time Stamp	F1	F2	F3	F4	F5	System	Memory	CPU	I/O	CPU
N1	8:29:49	4214	4289	4230	4264	4263	58	62	63	72	44
N2	6:09:23	3979	4046	4085	4060	4033	67	72	35	84	42
N3	7:20:22	4389	4403	4311	4286	4386	43	76	61	40	53
N4	6:53:08	5013	4928	5001	4981	5099	65	67	50	62	55
N5	6:30:09	3552	3635	3601	3591	3499	67	82	30	38	44
N6	6:01:46	4012	3970	3939	3891	3919	66	69	64	50	42

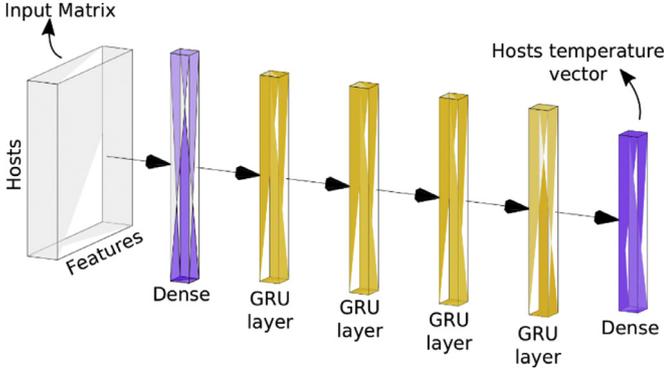


Fig. 4. Architecture of RNN to predict temperature of PMs.

directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. However, analyzing temperature characteristics also requires the exploitation of temporal workload and processing patterns, which can be done best with RNNs. However, RNNs face the problem of vanishing gradients which makes the network updates very slow, for which the research community has introduced GRUs. GRUs are a gating mechanism in recurrent neural networks, introduced in 2014 by Cho et al. (2014). To solve the vanishing gradient problem of a standard RNN, GRU uses, so-called, update gate and reset gate. These are two vectors which decide what information should be passed to the output. The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information, which is irrelevant to the prediction.

To train the neural network, we used traces¹ from Alibaba CDC, a large-scale ecommerce provider Li and Hu, 2019; Guo et al., 2019; Tian et al., 2019; Wu et al., 2019). Dataset extracted from traces on a cluster of 70 servers with 64-bit Ubuntu 18.04 Operating System, every server is equipped with the Intel® Core™ i7 9700k processor (No. of Cores 8, No. of Threads 8, Processor Base Frequency 3.60 GHz, Max Turbo Frequency 4.90 GHz, Cache 12 MB Intel® Smart Cache, Bus Speed 8 GT/s and TDP 95 W), 16 GB of RAM, 512 GB SSD storage and Graphics card GTX 2060. Hadoop MapReduce is installed on all the servers for processing to execute word count application. Every server hosts at least 15 virtual machines or nodes. A sample from the dataset is shown in Table 2. However, only utilization of various elements of the server is not sufficient for accurate prediction of temperature. Previous works show that Fan speeds also affect the server temperature (Capozzoli and Primiceri, 2015; Lazic et al., 2018). For our experiments, we have taken configuration with 5 fans, so we calculate Fan speed

from the different utilization metrics in the dataset. Fan speed is calculated using Eq. (19), which is multiplication of average utilization ($Avg_{utilization}$) and CPU temperature ($Temp$) (Capozzoli and Primiceri, 2015; Lazic et al., 2018; Liu et al., 2012; Dundu and Gao, 2014).

$$RPM_{Fan} = \alpha (Avg_{utilization} \times Temp) \quad (19)$$

where $\alpha = 1.5$ RPM/Degree Celsius

For 5 fans, we keep fan speed RPM_{Fan} , a random value between $[RPM_{Fan} - \frac{1}{2} \Delta RPM_{Fan}, RPM_{Fan} + \frac{1}{2} \Delta RPM_{Fan}]$, where ΔRPM_{Fan} is defined using Eq. (20).

$$\Delta RPM_{Fan} = \alpha (\Delta Avg_{utilization} \times Temp + Avg_{utilization} \times \Delta Temp) \quad (20)$$

As per the dataset, the precision of $Avg_{utilization}$ within 1% and precision of $Temp$ is within 1 Degree Celsius (°C). Using 1100 training and 100 test datapoints, our network is able to reach accuracy of 96.78%.

Using this deep learning-based prediction module, ThermoSim can also classify VMs and PMs based on thermal-aware technique even for CDCs where temperature sensors are unavailable.

3.5. Virtual machine classification and scheduling

Fig. 5 presents the process of virtual machine classification and allocation. Virtual machines are categorized into three classes in accordance with their thermal features: hot, warm, and cold. A 'cold' VM symbolizes that a VM may decrease the temperature of a PM if its temperature is greater than θ_{vl} . A 'hot' VM states that a VM may raise the temperature of a host if its temperature is greater than θ_{vh} . A 'warm' VM signifies the temperature of a PM will remain stable.

The thermal scheduler allocates VMs to the PM whose temperature is farthest away from its maximum threshold temperature. The scheduler also manages a waiting queue. It is the queue through which demand for a new VM is fulfilled. Each new request for VM is added at the end of waiting queue (Capozzoli and Primiceri, 2015).

The scheduler will remove that VM request from the waiting queue as per further requirement; then further scheduler queues the VMs into sub-queues based on the temperature variation of the host due to the VM i.e., value of ΔT_{vi} . If the temperature variation of host is greater than the high temperature threshold, then the VM is added to the hot queue; if the temperature variation of host is less than the low temperature threshold, then it is added to the cold queue, otherwise it is added to the warm queue (Chu et al., 2015; Capozzoli and Primiceri, 2015). For VM allocation, comparison is done: if the current temperature of a host is more than the high temperature threshold of a host, a cold VM from the cold queue will be allocated (Zhou and Wei, 2015). Subsequently, the host temperature lowers to the ordinary state and a VM will be allocated on the host to execute. When the temperature of a host is greater than θ_{ch} and VM from the cold queue is allocated to the

¹ Alibaba Cluster - <https://github.com/alibaba/clusterdata>.

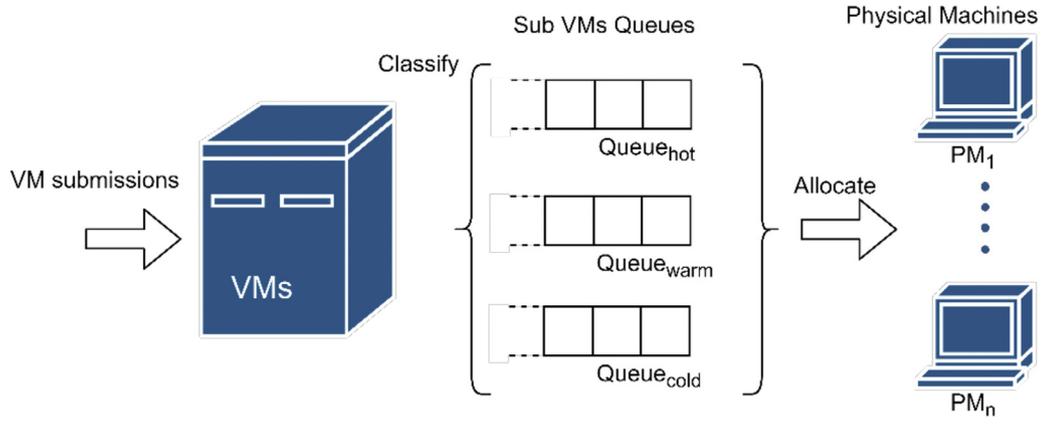


Fig. 5. Process of virtual machine classification and allocation.

host, the temperature then drops to below θ_{ch} . When the temperature of a host is less than θ_{cl} , a hot VM will be selected to execute. For cases that the temperature of a host is between θ_{cl} and θ_{ch} , the VM from warm queue will be executed. Algorithm 2 presents the thermal-aware approach, which contains two phases: 1) sorting and 2) resource scheduling.

Algorithm 2 Thermal-aware Approach for Resource Management.

```

1. Input: Number of VMs and number of available PMs
2. Output: Scheduling of VMs to the PMs
3. Start
4. Initialize all host list (Number of PMs)
5. Initialize all VMs list (Number of Resources (VMs) - vmlist)
6. do
7. {
8.   dequeue VM from queue
9.   calculate  $\Delta T_{vi}$ 
10.  if ( $\Delta T_{vi} > \theta_{vh}$ ) then
11.    enqueue the VM to  $Q_{Hot}$ 
12.  else if ( $\Delta T_{vi} < \theta_{vl}$ ) do
13.    enqueue the VM to  $Q_{cold}$ 
14.  else if ( $\theta_{vh} > \Delta T_{vi} > \theta_{vl}$ ) do
15.    enqueue the VM to  $Q_{warm}$ 
16. }
17. while !vmlist.empty()
18. if ( $T_{over} > \theta_{ch}$  and !  $Q_{cold}$ .empty()) then
19.   dequeue( $Q_{cold}$ );
20. else if (! $Q_{warm}$ .empty()) then
21.   dequeue( $Q_{warm}$ );
22. else
23.   dequeue( $Q_{Hot}$ );
24. if ( $T_{over} < \theta_{ch}$  & !  $Q_{Hot}$ .empty()) then
25.   dequeue( $Q_{Hot}$ );
26. else if (! $Q_{warm}$ .empty()) then
27.   dequeue( $Q_{warm}$ );
28. else dequeue( $Q_{cold}$ );
29. allocate VM to PM
30. End

```

The thermal-aware algorithm sorts the VMs based on their execution effect on PMs. Algorithm 2 executes hot VMs on cold nodes and cold VMs on hot nodes and tries to reduce energy consumption. The symbols used in the Algorithm 2 are defined in Table 3.

For calculating the value of θ_{vl} and θ_{vh} different types of VMs according to temperature variation of the host machine;

$$\theta_{vh} = T_{over} - T_{danger} \quad (21)$$

$$\theta_{vl} = \frac{1}{2} T_{normal} - T_{danger} \quad (22)$$

Several papers from the literature (Tsai and Chen, 2016; Chu et al., 2015; Zhou and Wei, 2015; Wu and Xu, 2010;

Table 3
Symbols and its description.

Symbol	Description
ΔT_{vi}	Temperature variation of host due to execution of current VM
T_{over}	Temperature of overheated host
T_{normal}	Temperature of normal host
T_{danger}	Temperature of overheating host
θ_{vh}	The low temperature threshold of ΔT_{vi}
θ_{vl}	The high temperature threshold of ΔT_{vi}
θ_{cl}	Low temperature threshold of host // normal host temperature
θ_{ch}	High temperature threshold of host //overheating host temperature

Ilager et al., 2019; Chaudhry et al., 2015) reported a different range of values for temperature:

- $T_{over} = 79$ °C. // overheated host temperature
- $T_{danger} = 70$ °C. //overheating host temperature
- $T_{normal} = 29$ °C. // normal host temperature

The main idea of Algorithm 2 is described below:

1. Initialize the all hosts (PMs) as available at time stamp to 0 i.e. host is available at the beginning of resource scheduling.
2. Calculate the temperature variation, before and after allocating the VM until queue is not empty.
3. Allocate VMs to sub-queues according to the temperature variation value of the host.
4. Dequeues VMs from their sub-queues according to the host state and allocates VM to PM.
5. Execute the allocated VM on PM.

Algorithm 2 maintains 3 sub-queues for VMs, classified into hot, warm and cold. The VMs are allocated to these queues based on their temperature models. Based on the queues and temperatures of hosts, we allocate hot VMs to cold hosts and cold ones to hot hosts. If hot or cold VM queues are empty, we next consider warm queues. This way the temperature remains balanced. The overall complexity of the algorithm is $O(n)$ where n is maximum of number of hosts and VMs.

3.6. Integration of all models

The utilization model is used in integration with the energy model. It provides quantified values for resource utilization for different computational elements like CPU, memory and disk. The energy model is used to calculate the energy consumption for different components classified broadly into computing and cooling. Both energy and utilization models are used for resource management of allocating tasks to VMs in ThermoSim as described in

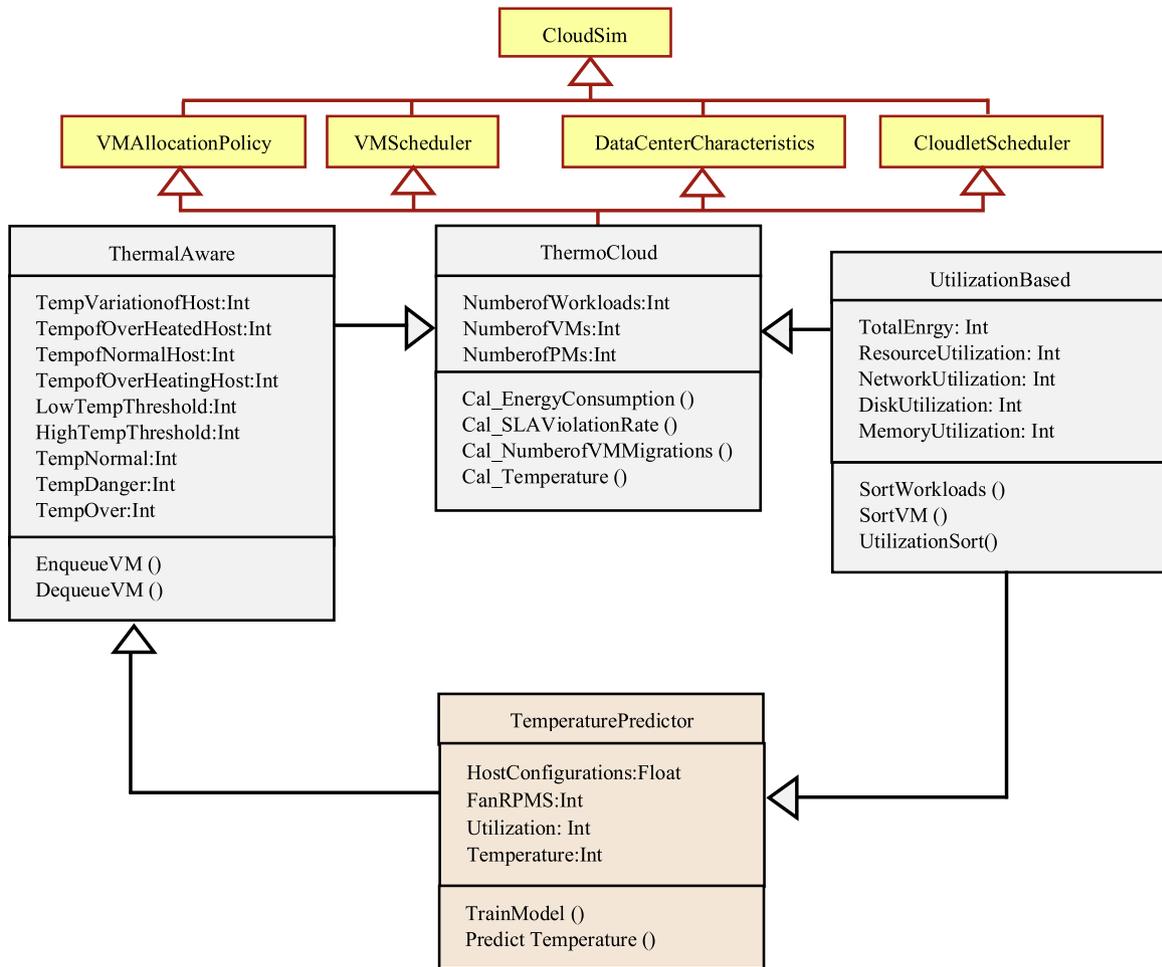


Fig. 6. ThermoSim framework is extending different classes from the CloudSim Toolkit.

Algorithm 1. Moreover, the thermal model using temperatures calculated by either Eq. (18) or the RNN based prediction module is used for thermal-aware approach for resource management of allocating VMs to PMs, as shown in Algorithm 2. We show use cases of both algorithms in performance evaluation, where we compare energy-based and thermal-based resource allocation policies as described in Section 4.

4. Performance evaluation

This section describes the performance evaluation, configuration details, type of workload and experimental results for validation of ThermoSim. The CloudSim toolkit (Calheiros et al., 2011) has been extended by developing a new ThermoSim framework to incorporate thermal parameters. Fig. 6 shows the class diagram where the ThermoSim framework is extending different classes (VMAllocationPolicy, VMScheduler, DataCenterCharacteristics and CloudletScheduler) from the CloudSim toolkit and developed four different classes (ThermoCloud, UtilizationBased, ThermalAware and TemperaturePredictor). ThermoCloud is main class, which interacts with classes of CloudSim toolkit and controls UtilizationBased class, ThermalAware class and TemperaturePredictor class. Fig. 7 shows the sequence diagram, which describes the interaction among various classes of ThermoSim framework during execution of workloads.

4.1. Configuration settings

We have conducted experiments on a machine with an Intel® Core™ i7-7820HQ Processor (8M Cache, 3.90 GHz), 16 GB RAM

and 2 TB of HDD running on 64-bit Windows OS. Our CDC comprises of 4 PMs with configuration (Cores = 4, CPU MIPS = 2000, RAM = 8 GB, Bandwidth = 1 Gbit/s) and 12 VMs with configuration (Core = 1, CPU MIPS = 500, RAM = 1 GB and Bandwidth = 100 Mbit/s). We have run the experiments for 10 times and the average results are reported. We have run the simulation for periods of 48 h and executed the scheduling algorithm after every 5-minute interval for dynamic consolidation of VMs. Virtual nodes are further divided into instances called Execution Components (ECs). Every EC contains their own cost of execution and it is measured with unit (C\$/EC time unit (Sec)). EC measures cost per time unit in Cloud dollars (C\$).

For the execution of workloads in our experiments, we have chosen varied computational settings on top of heterogeneous resources. The variety comes in the number of cores at the CPU-level, the page levels of the main memory, switches at the network level and disk space at the storage level (Calheiros et al., 2011; Grozev and Buyya, 2013; Lebre et al., 2015; Gill et al., 2019). Cores is the number of Processing Element's (PE) required by the Cloudlet. Table 4 shows the simulation parameters utilized in the various experiments undertaken by this research work, also as identified from the existing empirical studies and literature such as utilization model (Qinghui et al., 2008; Kouki and Ledoux, 2012), energy model (computing (Singh et al., 2016; Gill et al., 2019; Li et al., 2018; Balis et al., 2018; Lin et al., 2019, 2019) and cooling (Qinghui et al., 2008; Lazic et al., 2018; Möbius et al., 2014; Liu et al., 2012) and thermal-aware scheduling (Lazic et al., 2018; Ranganathan and Sharma, 2005; Möbius et al., 2014). Experimental setup incorporated CloudSim to produce and retrieve simulation results.

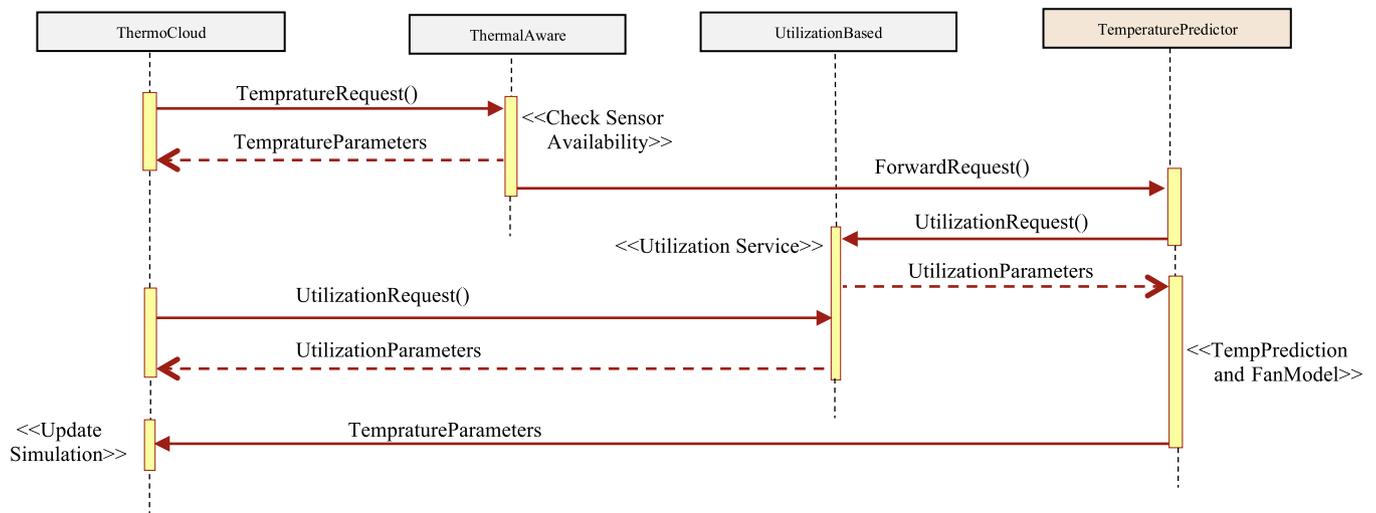


Fig. 7. Sequence diagram describes interaction among various classes of ThermoSim framework.

Table 4
Simulation parameters and their Values.

Parameter	Value
Number of VMs (nodes)	360
Number of Cloudlets (Workloads)	3000
Bandwidth	1000–3000 B/S
CPU MIPS	2000
Size of Cloud Workload	10000+ (10%–30%) MB
Number of PEs per machine	1
PE ratings	100–4000 MIPS
Cost per Cloud Workload	3 C\$–5 C\$
Memory Size	2048–12576 MB
File size	300 + (15%–40%) MB
Cloud Workload output size	300 + (15%–50%) MB
Initial Temperature	12–22 °C
Inlet Temperature	15–40 °C
Power Consumption by Processor	130 W–240 W
Power Consumption by Cooling Devices	400 W–900 W
Power Consumption by RAM	10 W–30 W
Power Consumption by Storage	35 W–110 W
Power Consumption by Network	70 W–180 W
Power Consumption by Extra Components	2 W–25 W

4.2. Workload

The ThermoSim framework uses PlanetLab² dataset (Park and Pai, 2006) and it is considered as a workload (Cloudlet). This dataset is a set of resource utilization traces from PlanetLab VMs collected during 10 random days in March and April 2011. To find the experiment statistics, 500–3000 different workloads are executed. We selected the Poisson Distribution (Amvrosiadis et al., 2018) for workload submission in this research work due to following reasons: 1) evaluating the performance of workload execution for specific interval of time and 2) every workload is independent of all other workloads (number of workloads are arriving in first hour is independent of the number of workloads arriving in any other hour).

4.3. Validation of ThermoSim

We have validated the proposed ThermoSim framework using an existing Thermal-aware Simulator, called **ThaS** (Mhedheb et al.,

2013), which implements the thermal-aware scheduling policy using CloudSim toolkit (Calheiros et al., 2011).

4.3.1. Baseline simulator

ThaS (Mhedheb et al., 2013) implements an interface between the hypervisor and the virtual machine in CDC. They have replaced the CloudSim's VMScheduler class which results in their scheduling behavior to be highly data-dependent of the existing class inputs. However, ThaS lacks thermal characteristic studies by which they are not able to find the best host location at the time of VM migration. Moreover, their model is restricted in terms of analyzing the CPU and memory loads and the number of VM migrations which significantly impacts the performance of the system in terms of its latency and other QoS parameters.

In ThermoSim, we have implemented the thermal-behavior by developing three new classes (ThermalAware, UtilizationBased and TemperaturePredictor), which further extends the four classes (DataCenterCharacteristics, VMScheduler, VMAllocationPolicy and CloudletScheduler) of the CloudSim toolkit (Calheiros et al., 2011) through a coordinator class, *ThermoCloud*, as shown in Fig. 6. The ThermalAware class takes temperature input from the TemperaturePredictor class when temperature information of the cloud hosts is not available. The TemperaturePredictor class uses utilization metrics from the UtilizationBased class and applies Fan-RPM models if such information is unavailable. We provided standard error bars for every graph to show the variation in the experimental results.

Experimental Results: Fig. 8 (a) shows the variation of memory usage with varying numbers of workloads. ThermoSim saves 12.6% memory as compared to ThaS, which increases further to 14.74% when using the prediction module. Fig. 8 (b) shows the variation of time against the number of workloads. ThermoSim saves 17.7% of time as compared to ThaS, which increases further to 23.64% when using the prediction module. Fig. 8 (c) shows the variation of cost with number of workloads which is measured in Cloud Dollars (C\$) and ThermoSim saves 15.5% cost as compared to ThaS, which increases further to 20.45% when using the prediction module. Fig. 8 (d) shows variation of energy with the number of workloads and ThermoSim saves 10.5% energy as compared to ThaS, which increases further to 14.13% when using the prediction module.

4.3.2. Prediction accuracy

We have evaluated the prediction accuracy for both ThermoSim and ThaS to prove the novelty of proposed framework with varia-

² PlanetLab - [https://wikitech.wikimedia.org/wiki/Analytics/Archive/Data/\(?PMU?\)Pagecounts-raw](https://wikitech.wikimedia.org/wiki/Analytics/Archive/Data/(?PMU?)Pagecounts-raw).

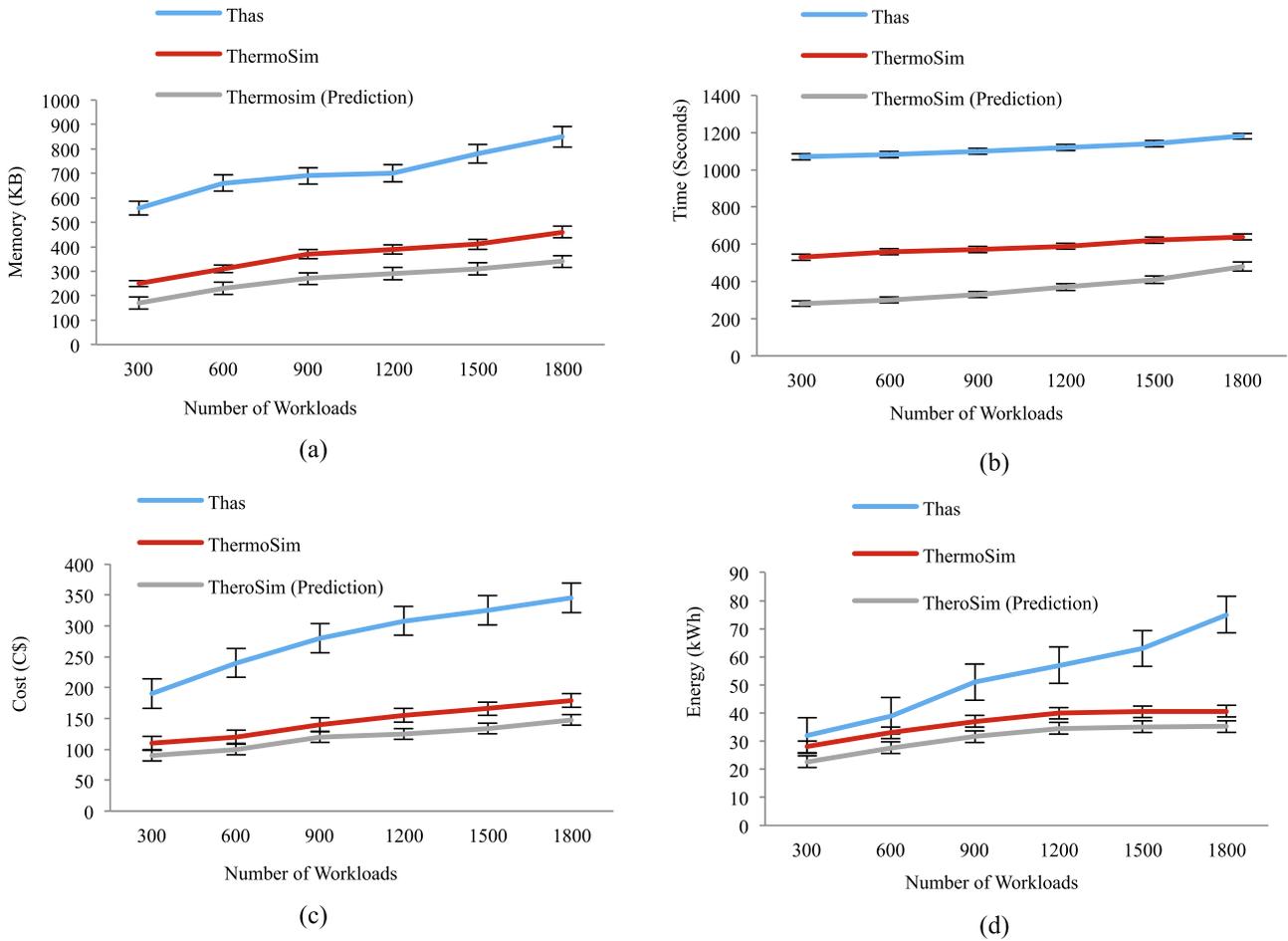


Fig. 8. Performance comparison of ThermoSim and Thas with different Number of Workloads: (a) Memory (b) Time, (c) Cost, (d) Energy.

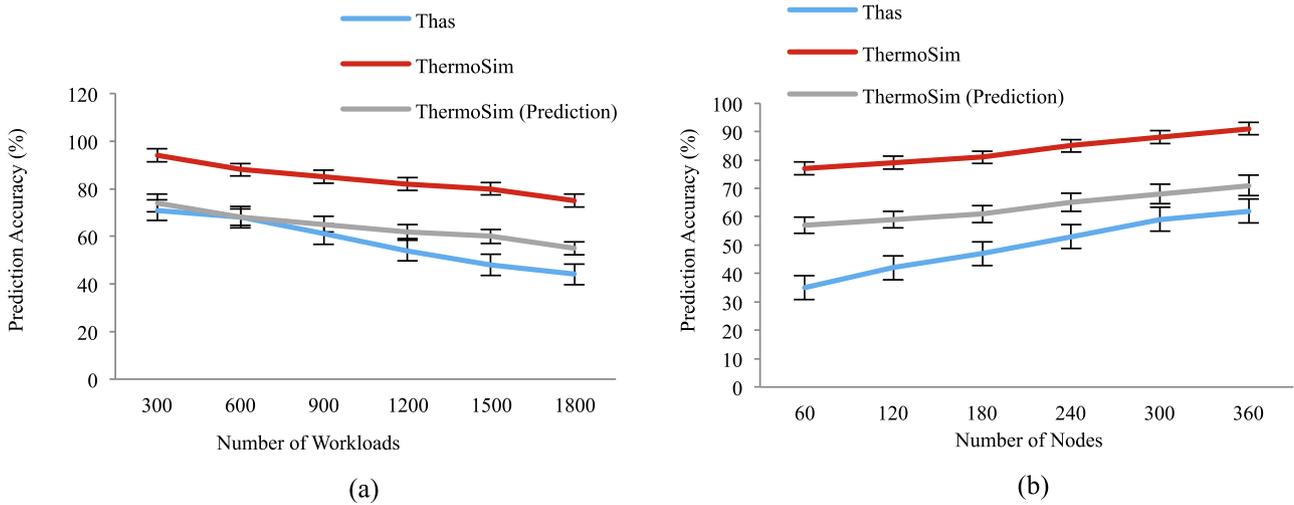


Fig. 9. Performance comparison of ThermoSim and Thas in terms of Prediction Accuracy for Resource Requirement: (a) Different Number of Workloads and (b) Different Number of Nodes.

tion of workloads and nodes. *Prediction Accuracy* is defined as the ratio of the number of correct predictions in the experiment to the total number of predictions in the experiment for thermal, energy and utilization.

Fig. 9 shows the comparison of ThermoSim and Thas based on prediction accuracy for resource requirement. Fig. 9(a) shows the performance comparison of ThermoSim and Thas with different number of workloads and the value of prediction accuracy is decreasing with the increase in the number of workloads but Ther-

moSim performs better than Thas (both with and without the prediction module). Fig. 9(b) shows the performance comparison of ThermoSim and Thas with different number of nodes and the value of prediction accuracy is increasing with the increase in the number of nodes but ThermoSim performs better than Thas (both with and without the prediction module).

Fig. 10 shows the comparison of ThermoSim and Thas based on prediction accuracy for temperature. Here too, the prediction accuracy decreases with increasing number of workloads and in-

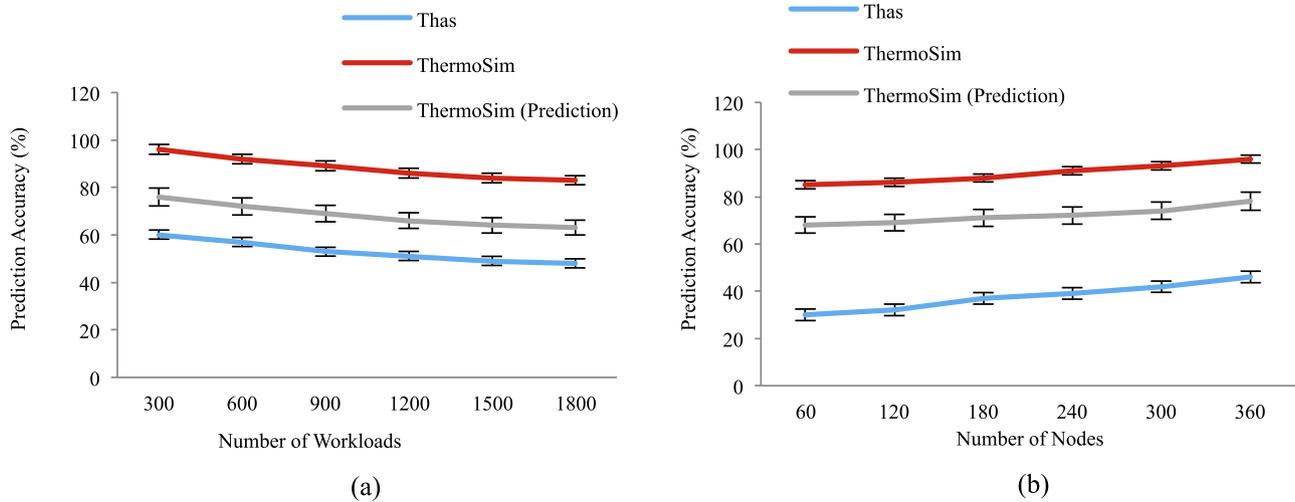


Fig. 10. Performance comparison of ThermoSim and Thas in terms of Prediction Accuracy for Temperature: (a) Different Number of Workloads and (b) Different Number of Nodes.

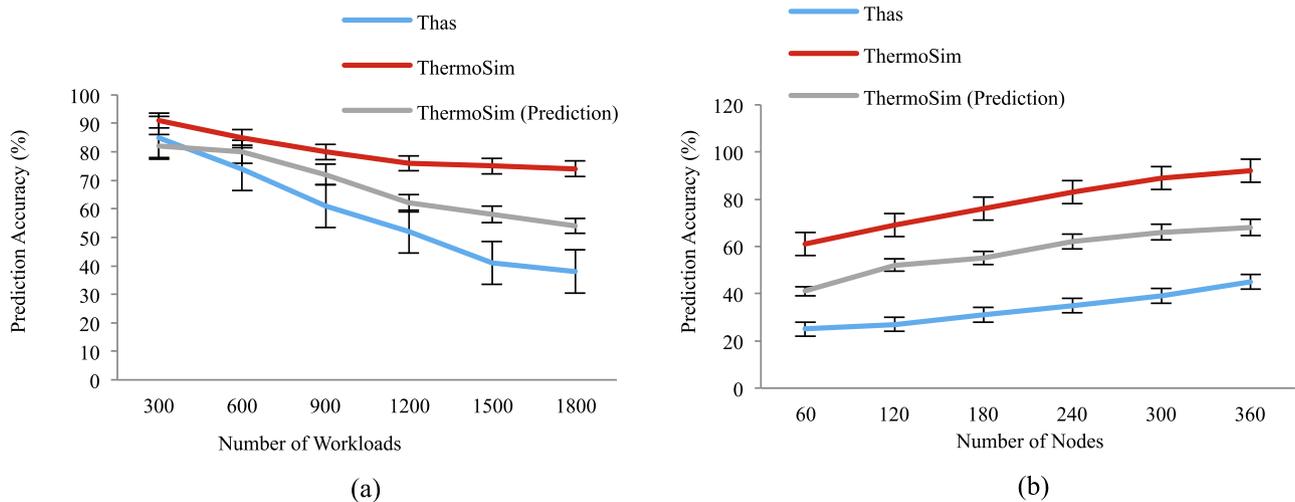


Fig. 11. Performance comparison of ThermoSim and Thas in terms of Prediction Accuracy for Energy Consumption: a) Different Number of Workloads and b) Different Number of Nodes.

creases with increasing number of nodes. Fig. 10 shows that the ThermoSim (both with and without prediction module) gives better performance as compared to Thas.

Fig. 11 shows the comparison of ThermoSim and Thas based on prediction accuracy for energy consumption. Fig. 11(a) shows that the prediction accuracy falls drastically for Thas from 85% to 38% as number of workloads increases from 300 to 1800. Accuracy increases with number of nodes for Thas and ThermoSim (with and without the prediction module) as shown in Fig. 11(b).

Fig. 12 shows the comparison of ThermoSim and Thas based on prediction accuracy for resource utilization. Again, ThermoSim has higher accuracy compared to Thas. However, when the prediction module is used, the accuracy falls drastically with increasing number of workloads due to higher variance in memory usage characteristics at large number of workloads leading to thermal throttling in systems, hence reducing prediction performance (Lin et al., 2007).

Fig. 13 shows the comparison of ThermoSim and Thas based on prediction accuracy for memory utilization. Again, accuracy for ThermoSim (both with and without prediction module) is higher compared to Thas.

Fig. 14 shows the comparison of ThermoSim and Thas based on prediction accuracy for disk utilization. Again, accuracy for Ther-

moSim (both with and without prediction module) is higher as compared to Thas. However, disk utilization prediction accuracy is low when using prediction modules due to high variance and task heterogeneity in the system (Longford, 2010).

Fig. 15 shows the comparison of ThermoSim and Thas based on prediction accuracy for network utilization. As shown, the network utilization prediction accuracy decreases with increasing number of workloads and increases with increasing number of nodes.

4.3.3. Analysis of results

The ThermoSim framework is capable of scheduling the VMs independently. Experimental results show that ThermoSim has better performance compared to Thas as Thas implements thermal-aware scheduling policy inside the VMScheduler class of CloudSim toolkit (Calheiros et al., 2011), and as a result its scheduling behavior is still dependent on the VMScheduler class - which increases data-dependency. There are two other important reasons for better performance of ThermoSim as compared to Thas: 1) ThermoSim is effective in locating and scheduling the energy efficient resources dynamically using CRUZE (Gill et al., 2019) and 2) ThermoSim performs scaling operations sharply. The prediction accuracy reduces when we use the prediction module as it does not consider various room cooling methods like CRAC, however due to much simpler

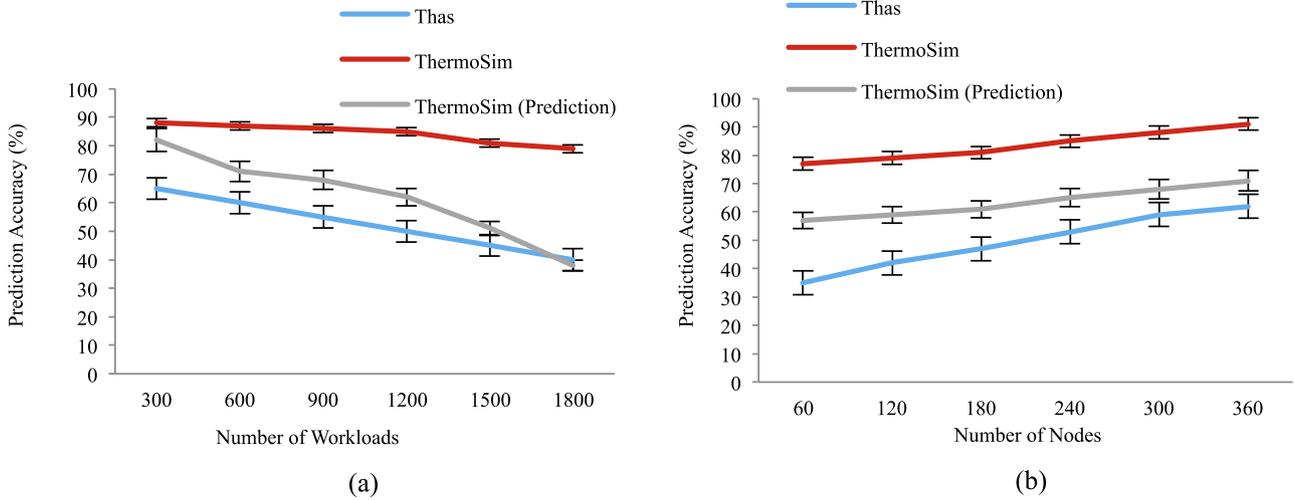


Fig. 12. Performance comparison of ThermoSim and Thas in terms of Prediction Accuracy for Resource Utilization: (a) Different Number of Workloads and (b) Different Number of Nodes.

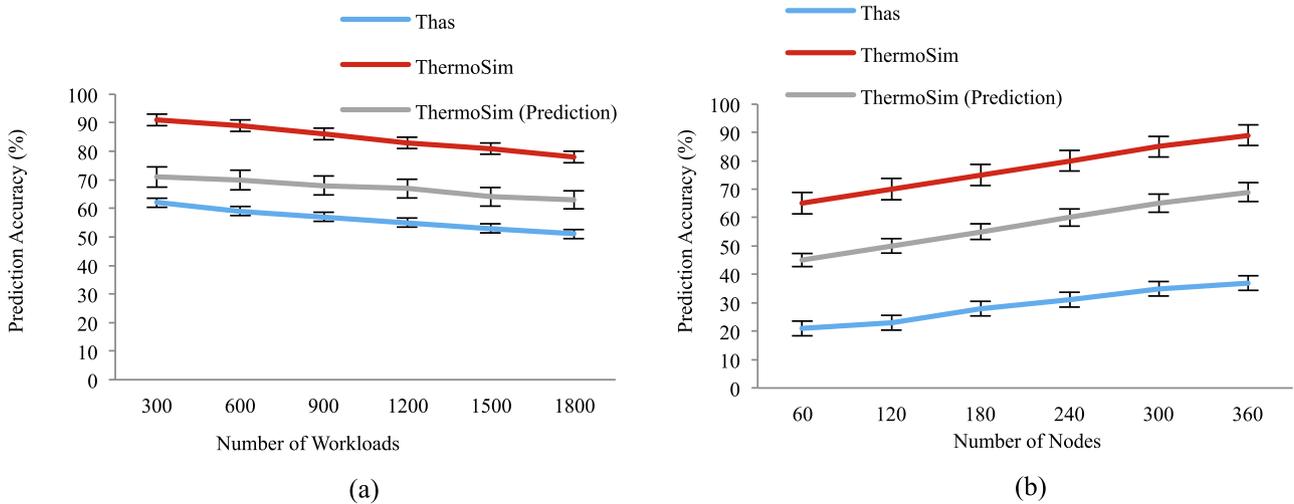


Fig. 13. Performance comparison of ThermoSim and Thas in terms of Prediction Accuracy for Memory Utilization:(a) Different Number of Workloads and (b) Different Number of Nodes.

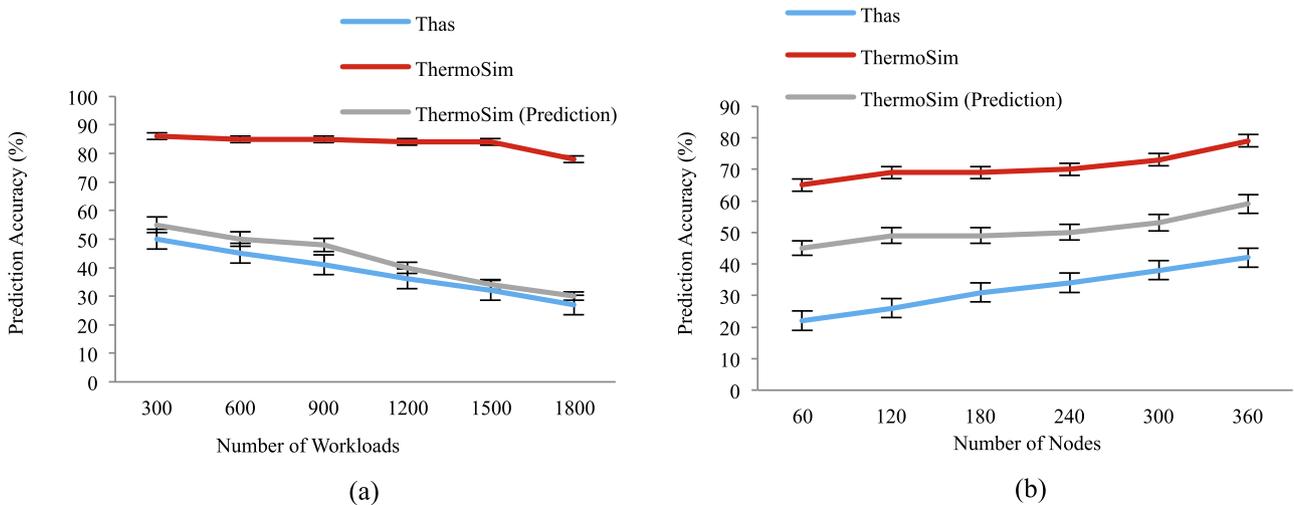


Fig. 14. Performance comparison of ThermoSim and Thas in terms of Prediction Accuracy for Disk Utilization: (a) Different Number of Workloads and (b) Different Number of Nodes.

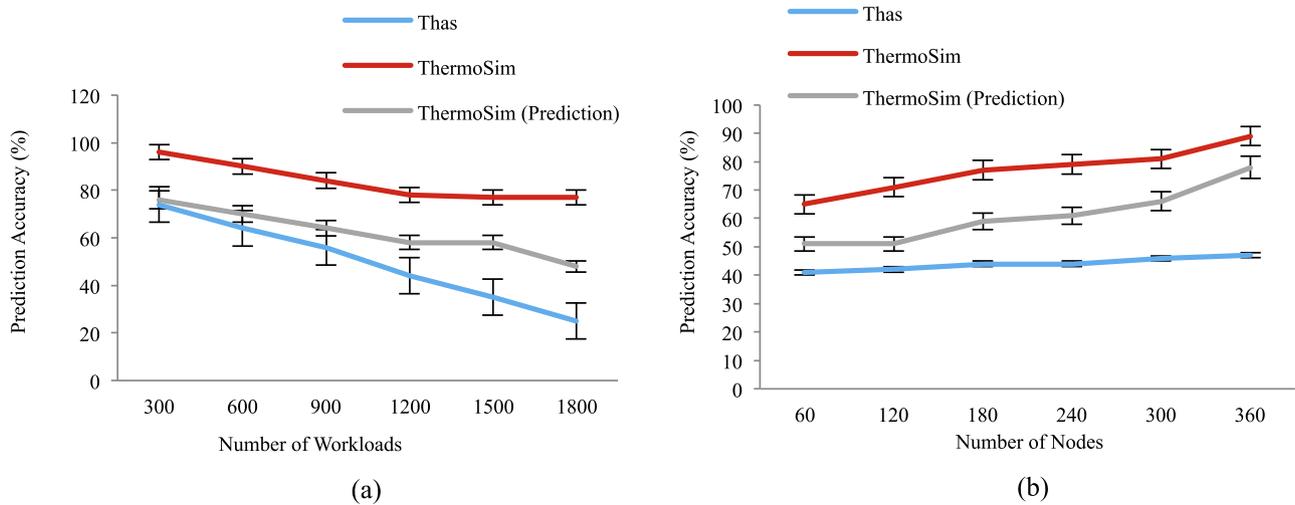


Fig. 15. Performance comparison of ThermoSim and Thas in terms of Prediction Accuracy for Network Utilization: (a) Different Number of Workloads and (b) Different Number of Nodes.

implementation and lower computational requirements of the prediction module. **Note:** The detailed description all the metrics are given in our previous work (Singh et al., 2016) (Gill et al., 2019).

4.4. Case studies using ThermoSim

We have presented two case studies to test the performance of three different energy-aware and thermal-aware resource management techniques using both the proposed ThermoSim framework and the existing framework Thas. The energy consumption shown here is based on average energies calculated by the linear and non-linear models. For brevity and fair comparison, we only include results without using the prediction module and using the Thermal model instead. The results when using the RNN are nearly between the ThermoSim and Thas results, as its prediction accuracies are lower than when Thermal model is used. This experiment used 1200 workloads. We have tested the performance of ThermoSim and Thas using QoS parameters such as energy consumption, SLA Violation Rate (SVR), number of VM migrations and temperature as shown in Figs. 16 and 17.

4.4.1. Case study 1: energy-aware resource management

In this section, first case study has been presented to test the performance of three different energy-aware resource management techniques (FCFS, DVFS and SOCCER) using both the proposed ThermoSim framework and the existing framework Thas (Mhedheb et al., 2013). First Come First Serve (FCFS) based energy-aware resource management technique schedules the resources for execution of homogeneous workloads using FCFS-based scheduling algorithm. Dynamic Voltage and Frequency Scaling (DVFS) (Chia-Ming et al., 2014) is an energy optimization approach, which adjusts the frequency settings of the computing devices to optimize scheduling of resources. SOCCER (Singh et al., 2016) is an energy-aware autonomic resource scheduling approach, which executes the heterogeneous cloud workloads using the IBM autonomic model. Fig. 16 shows the variation of QoS parameters for different energy-aware resource management techniques.

The value of QoS parameters increases as the number of nodes increases. Fig. 16 (a) shows SOCCER has 13.12% less SLA violation rate in ThermoSim compared to Thas, similarly in FCFS and DVFS 12.45% and 14.98% improvement respectively. SOCCER performed better than DVFS and FCFS in both ThermoSim and Thas framework as it executes workloads based on signed SLA between user and provider. Fig. 16 (b) shows SOCCER consumes 11.23% less energy in ThermoSim compared to Thas, similarly in FCFS and DVFS

8.35% and 9.15% improvement respectively. SOCCER performs better than DVFS and FCFS in both ThermoSim and Thas framework as it adjusts resource utilization at runtime.

Fig. 16 (c) shows SOCCER has 10.45% fewer VM migrations in ThermoSim compared to Thas, similarly in FCFS and DVFS 5.15% and 5.91% improvement respectively. SOCCER perform better than DVFS and FCFS in both ThermoSim and Thas framework as it is capable of performing resource consolidation dynamically. Fig. 16 (d) shows SOCCER offers 12.52% lower temperatures in ThermoSim compared to Thas, similarly in FCFS and DVFS 8.75% and 10.66% improvement respectively. SOCCER performs better than DVFS and FCFS in both ThermoSim and Thas framework as it shuts down the idle resources automatically during the execution of workloads.

4.4.2. Case study 2: thermal-aware resource management

In this section, second case study has been presented to test the performance of three different thermal-aware resource management techniques (DTM, ETAS and GTARA) using both the proposed ThermoSim framework and the existing framework Thas (Mhedheb et al., 2013). Energy and Thermal-Aware Scheduling (ETAS) algorithm (llager et al., 2019) that dynamically consolidates VMs to minimize the overall energy consumption while proactively preventing hotspots. Dynamic Thermal Management (DTM) technique (Kim et al., 2019) exploits external computing resources (idle servers) adaptively as well as internal computing resources (free cores of CPU in the server) available in heterogeneous data centers. Game-based Thermal-Aware Resource Allocation (GTARA) strategy (Akbar et al., 2019) decreases the imbalance within the CDC by using the concept of cooperative game theory with a Nash-bargaining to assign resources based on thermal profile. Fig. 17 shows the variation of QoS parameters for different thermal-aware resource management techniques.

The value of QoS parameters increases as the number of nodes increases. Fig. 17 (a) shows GTARA has 15.32% less SLA violation rate in ThermoSim compared to Thas, similarly in DTM and ETAS 16.66% and 13.72% improvement respectively. GTARA performs better than ETAS and DTM in both ThermoSim and Thas framework as it reduces the energy consumption significantly with low VM migrations while not violating the SLAs. Fig. 17 (b) shows GTARA consumes 9.73% less energy in ThermoSim compared to Thas, similarly in DTM and ETAS 11.93% and 12.65% improvement respectively. GTARA performs better than DTM and ETAS in both ThermoSim and Thas framework as it considers ambient effect of surrounding nodes is considered while assigning workloads to the computing nodes.

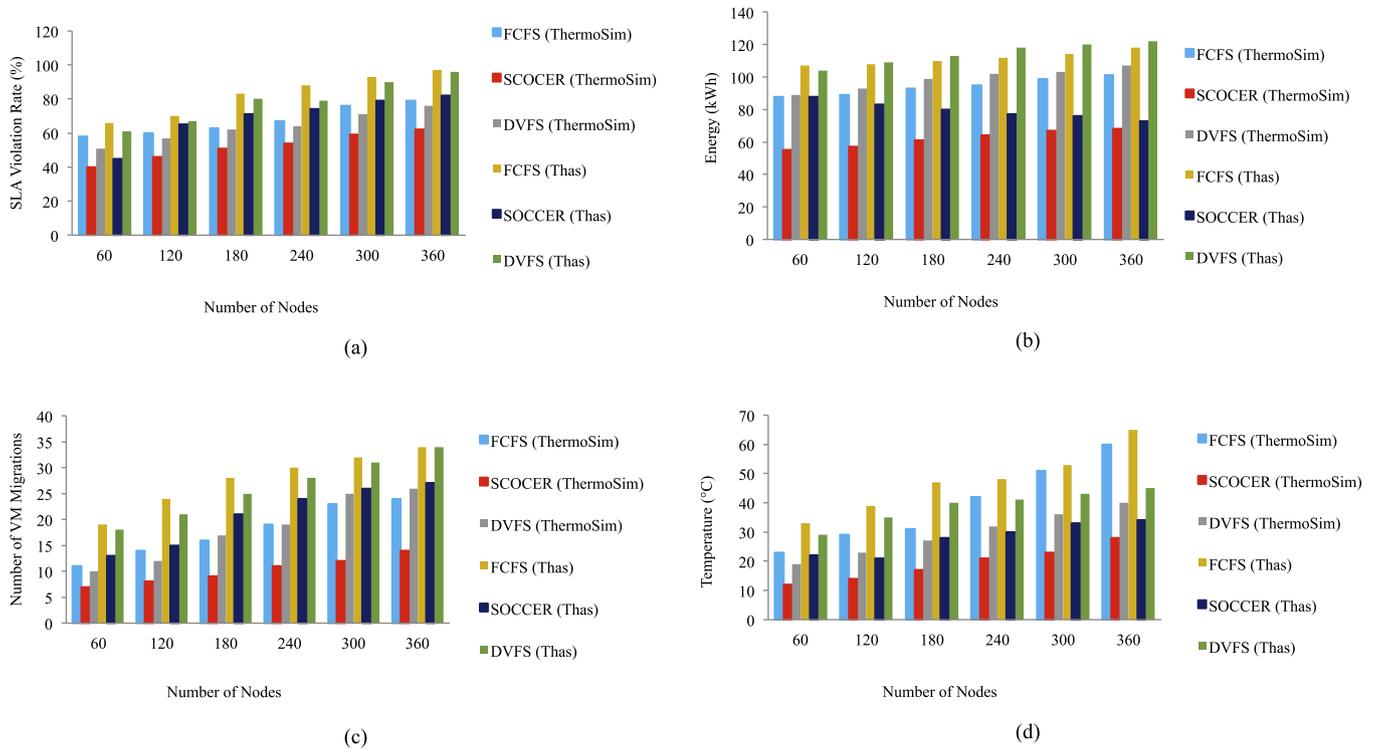


Fig. 16. Performance of different energy-aware scheduling algorithms: (a) SLA Violation Rate (b) Energy Consumption, (c) Number of VM Migrations, (d) Temperature.

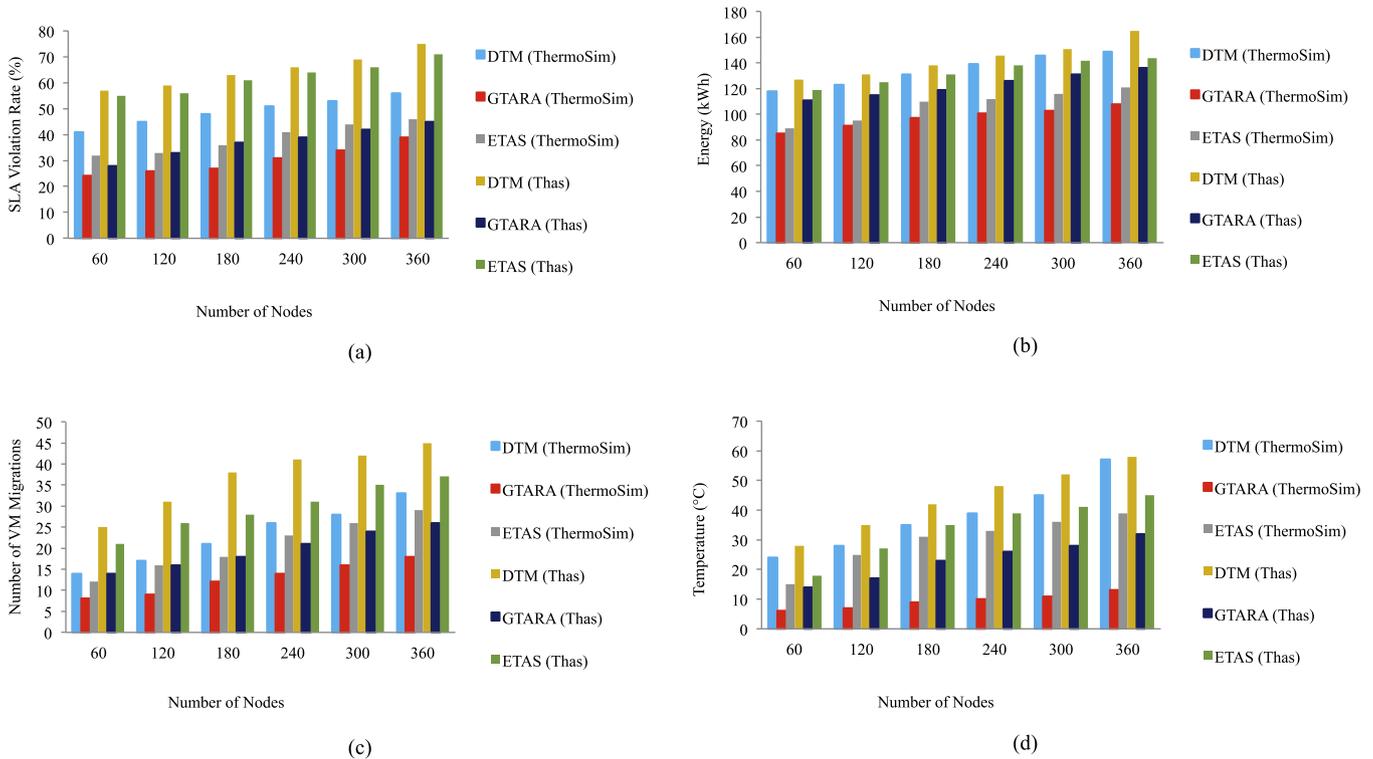


Fig. 17. Performance of different thermal-aware scheduling algorithms: (a) SLA Violation Rate (b) Energy Consumption, (c) Number of VM Migrations, (d) Temperature.

Fig. 17 (c) shows GTARA has 10.87% fewer VM migrations in ThermoSim compared to Thas, similarly in DTM and ETAS 6.46% and 7.75% improvement respectively. GTARA perform better than ETAS and DTM in both ThermoSim and Thas framework as it can improve thermal balance and avoid hotspots dynamically by using cooperative game theory with a Nash-bargaining solution.

Fig. 17 (d) shows GTARA offers 18.65% lower temperatures in ThermoSim compared to Thas, similarly in DTM and ETAS 16.95% and 17.75% improvement respectively. GTARA perform better than DTM and ETAS in both ThermoSim and Thas framework because workloads are assigned to the nodes based on their thermal profiles.

4.5. Discussions

The experimental results demonstrate that ThermoSim is performing better than Thas in terms of different performance parameters such as memory, energy, temperature and cost. Further, it is clearly noted that the prediction accuracy of ThermoSim is better than Thas with the variation of number of workloads and nodes. The more holistic approach of ThermoSim consideration of diverse parameters like SLA violation rate, cost, number of VM migrations and energy independently for each VM allows it to surpass Thas in performance. As per the results of case study 1 using ThermoSim, SOCCER has better performance as compared to DVFS and FCFS using different QoS parameters for energy-aware techniques, while GTARA perform better than ETAS and DTM for thermal-aware techniques as shown in case study 2.

5. Summary and conclusions

In this paper, we proposed a framework called ThermoSim for the simulation and modeling of thermal-aware resource management for cloud computing environments. ThermoSim uses a lightweight RNN-based deep learning model for thermal-predictor for efficient and low overhead resource management in resource constrained cloud environments. We have validated the proposed ThermoSim framework using three well-known energy-aware and thermal-aware resource scheduling techniques by testing the performance of four QoS parameters: energy consumption, SLA violation rate, number of VM migrations and temperature, with different number of resources. We have validated the proposed ThermoSim framework against existing thermal-aware simulator (Thas). Finally, relationship between theory and practice is very important. Benchmarking is an important starting point, which may try to relate the holistic aspects studied in our simulation in real-world practice. This may lead to additional improvements of the theoretical basis.

5.1. Future research directions and open challenges

Although the ThermoSim framework is capable of simulating and modeling the thermal-aware characteristics of cloud data centers, it can be further enhanced in a larger scope under the following aspects.

1. *High Energy Demand in Cooling Servers*: High power consumption leads to the creation of hot spots and increases in server temperature. Thus, there is a need to study the trade-off between cooling energy and computing energy and its effect on temperature.
2. *Peak Temperature among Servers*: Temperature is another important parameter for both physical servers and virtualization solutions. Variance in the on-chip temperature and the resultant occurrence of hot spots degrades the performance of processors, increases the energy consumption. Thermal management strategies are required to uniformly distribute the temperature.
3. *High Level of Power Consumption by the Servers*: The non-energy aware scheduling techniques lead to increase in power consumption among the servers which degrades the server's reliability and performance in terms of availability and scalability.
4. *Validation of ThermoSim in the Large-scale Cloud Data Center*: The proposed ThermoSim framework will be implemented in a real cloud environment to validate the availability of the proposed models using Hadoop based Cloud Cluster and thermal sensors.
5. *Automation of ThermoSim using Artificial Intelligence (AI)*: To build a framework/benchmark that automatically classifies

applications/workloads according to their temperature profile using AI (Gill et al., 2019).

6. *Fog and Edge Computing Environments*: Further, ThermoSim can be extended towards fog/edge computing scenarios with heterogeneous hardware, and stronger energy constraints (Gill et al., 2019).
7. *Container-based Deployment*: In future, virtualization technology (e.g., VMs) used in ThermoSim can be replaced with Docker-based containers to improve the performance for CPU-intensive applications. Ease of use of containers (especially quick restarts) can reduce the execution time of workloads and improves energy efficiency (Zhou et al., 2020). Containers also provide a lightweight environment for the deployment applications because they are stand-alone, self-contained units that package software and its dependencies together. Further, container-based deployment is more effective than VMs because containers have small memory footprint and consume a very small amount of resources (Gill et al., 2019).
8. *Reliability and Security*: There is a need to investigate the interlink between cloud thermo properties and dependable provision of computation, covering scheduling, resource allocation, consolidation, execution, cloudlet among the others etc. By dependability, we refer to the extent to which the cloud environment can continue to maintain its reliability, security and performance while dynamically optimizing for energy and honoring thermo-properties (Gill et al., 2019).
9. *Dynamic Monitoring*: There is a need to develop priority tools and repositories for monitoring and collecting behavioural information about the dynamic management of thermo properties and dependability. These repositories can be mined to inform the design of more dependable thermo-aware policies.

Datasets

The datasets used for this research work are available at:

Alibaba Cluster - <https://github.com/alibaba/clusterdata>
PlanetLab - <https://wikitech.wikimedia.org/wiki/Analytics/Archive/Data/Pagecounts-raw>

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Sukhpal Singh Gill: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Validation, Writing - original draft, Writing - review & editing. **Shreshth Tuli**: Conceptualization, Data curation, Methodology, Software, Validation, Writing - review & editing. **Adel Nadjaran Toosi**: Writing - review & editing, Data curation, Supervision. **Felix Cuadrado**: Formal analysis, Software, Writing - review & editing. **Peter Garraghan**: Writing - original draft, Formal analysis, Supervision, Methodology, Writing - review & editing. **Rami Bahsoon**: Data curation, Writing - review & editing. **Hanan Lutfiyya**: Visualization, Writing - review & editing. **Rizos Sakellariou**: Formal analysis, Writing - review & editing. **Omer Rana**: Conceptualization, Supervision, Writing - review & editing. **Schahram Dustdar**: Visualization, Writing - review & editing. **Rajkumar Buyya**: Conceptualization, Writing - original draft, Supervision, Formal analysis, Visualization, Methodology, Writing - review & editing.

Acknowledgements

We would like to thank Joseph Richardson (Lancaster University, UK), Shashikant Ilager (University of Melbourne, Australia), Shikhar Tuli (IIT Delhi, India) and Dominic Lindsay (Lancaster University, UK) for their feedback to improve the quality of the paper. We would like to thank the editors, area editor and anonymous reviewers for their valuable comments and suggestions to help and improve our research paper. An initial investigation on this work was carried out at Melbourne CLOUDS Lab, which was supported by Melbourne-Chindia Cloud Computing (MC3) Research Network.

References

- Akbar, S., Malik, S.U.R., Samee, U.K., Raymond, C., Adeel, A., Naveed, A., 2019. A Game-based thermal-aware resource allocation strategy for data centers. *IEEE Transact. Cloud Comput.*
- Amvrosiadis, G., Park, J.W., Gregory, R.G., Gibson, G.A., Baseman, E., DeBardeleben, N., 2018. On the diversity of cluster workloads and its impact on research results. 2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18) 533–546.
- Bakker, A., Huijsing, J., 2013. High-Accuracy CMOS Smart Temperature Sensors, 595. Springer Science & Business Media.
- Balis, B., Brzoza-Woch, R., Bubak, M., Kasztelnik, M., Kwolek, B., Nawrocki, P., Nowakowski, P., Szydło, T., Zielinski, K., 2018. Holistic approach to management of IT infrastructure for environmental monitoring and decision support systems with urgent computing capabilities. *Future Gener. Comput. Syst.* 79, 128–153.
- Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F.D., Buyya, R., 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software* 41 (1), 23–50.
- Capozzoli, A., Primiceri, G., 2015. Cooling systems in data centers: state of art and emerging technologies. *Energy Procedia* 83, 484–493.
- Chaudhry, M.T., Ling, T.C., Manzoor, A., Hussain, S.A., Kim, J., 2015. Thermal-aware scheduling in green data centers. *ACM Comput. Surv.* 47 (3), 1–39.
- Chen, Y.-J., Horng, G.-J., Li, J.-H., Cheng, S.-T., 2017. Using thermal-aware VM migration mechanism for high-availability cloud computing. *Wirel. Pers. Commun.* 97 (1), 1475–1502.
- Chia-Ming, W., Chang, R., Chan, H., 2014. A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters. *Future Gener. Comput. Syst.* 37, 141–147.
- Chu, H.-H., Kao, Y.-C., Chen, Y.-S., 2015. Adaptive thermal-aware task scheduling for multi-core systems. *J. Syst. Softw.* 99, 155–174.
- Dey, R., Fathi, M.S., 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In: 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). *IEEE*, pp. 1597–1600.
- Construction materials and structures, 2014. In: Dundu, M., Gao, X. (Eds.), *Proceedings of the first international conference on construction materials and structures*. IOS Press.
- Fu, L., Wan, J., Yang, J., Cao, D., Zhang, G., 2017. Dynamic thermal and IT resource management strategies for data center energy minimization. *J. Cloud Comput.* 6 (1), 1–16.
- Fu, L., Wan, J., Liu, T., Gui, X., Zhang, R., 2017. A temperature-aware resource management algorithm for holistic energy minimization in data centers. In: 2017 2nd Workshop on Recent Trends in Telecommunications Research (RTTR). *IEEE*, pp. 1–5.
- Gill, S.S., Garraghan, P., Stankovski, V., Casale, G., Ghosh, S.K., Thulasiram, R.K., Ramamohanarao, K., Buyya, R., 2019. Holistic resource management for sustainable and reliable cloud computing: an innovative solution to global challenge. *J. Syst. Softw. Elsevier* 155, 104–129.
- Gill, S.S., Tuli, S., Xu, M., Singh, I., Singh, K.V., Lindsay, D., Tuli, S., et al., 2019. Transformative effects of IoT, Blockchain and artificial intelligence on cloud computing: evolution, vision, trends and open challenges. *Internet Things* 8, 100118.
- Grozev, N., Buyya, R., 2013. Performance modelling and simulation of three-tier applications in cloud and multi-cloud environments. *Comput. J.* 58 (1), 1–22.
- Guo, J., Chang, Z., Wang, S., Ding, H., Feng, Y., Mao, L., Bao, Y., 2019. Who limits the resource efficiency of my datacenter: an analysis of Alibaba datacenter traces. In: *Proceedings of the International Symposium on Quality of Service*, pp. 1–10.
- Ilager, S., Ramamohanarao, K., Buyya, R., 2019. ETAS: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with proactive hotspot mitigation. *Concurrency Comput.* 31 (17), e5221.
- Khaleel, M.I., 2018. Load balancing and thermal-aware in geo-distributed cloud data centers based on Vlans. *Sci. J. Uni. Zakho* 6 (3), 112–117.
- Kim, Y.G., Kim, J.I., Choi, S.H., Kim, S.Y., Chung, S.W., 2019. Temperature-aware adaptive VM allocation in heterogeneous data centers. In: 2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). *IEEE*, pp. 1–6.
- Kouki, Y., Ledoux, T., 2012. Sla-driven capacity planning for cloud applications. In: *Cloud Computing Technology and Science (CloudCom)*, 2012 IEEE 4th International Conference on. *IEEE*, pp. 135–140.
- Kumar, M.R.V., Raghunathan, S., 2016. Heterogeneity and thermal aware adaptive heuristics for energy efficient consolidation of virtual machines in infrastructure clouds. *J. Comput. Syst. Sci.* 82 (2), 191–212.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv:1406.1078.
- Lazic, N., Boutillier, C., Lu, T., Wong, E., Roy, B., Ryu, M.K., Imwalle, G., 2018. Data center cooling using model-predictive control. *Adv. Neural Inf. Proc. Syst.* 3814–3823.
- Lebre, A., Legrand, A., Suter, F., Veyre, P., 2015. Adding storage simulation capacities to the simgrid toolkit: concepts, models, and api. In: 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. *IEEE*, pp. 251–260.
- Lee, E.K., Viswanathan, H., Pompili, D., 2017. Proactive thermal-aware resource management in virtualized HPC cloud datacenters. *IEEE Transact. Cloud Comput.* 5 (2), 234–248.
- Li, F., Hu, B., 2019. DeepJS: job scheduling based on deep reinforcement learning in cloud data center. In: *Proceedings of the 2019 4th International Conference on Big Data and Computing*, pp. 48–53.
- Li, X., Jiang, X., Garraghan, P., Wu, Z., 2018. Holistic energy and failure aware workload scheduling in cloud datacenters. *Future Gener. Comput. Syst.* 78, 887–900.
- Lin, J., Zheng, H., Zhu, Z., David, H., Zhang, Z., 2007. Thermal modeling and management of DRAM memory systems. In: *Proceedings of the 34th annual international symposium on Computer architecture*, pp. 312–322.
- Lin, W., Wu, G., Wang, X., Li, K., 2019. An artificial neural network approach to power consumption model construction for servers in cloud data centers. *IEEE Transact. Sustain. Comput.*
- Lin, W., Peng, G., Bian, X., Xu, S., Chang, V., Li, Y., 2019. Scheduling algorithms for heterogeneous cloud environment: main resource load balancing algorithm and time balancing algorithm. *J. Grid Comput.* 17 (4), 699–726.
- Lin, W., Wu, W., He, L., 2019. An on-line virtual machine consolidation strategy for dual improvement in performance and energy conservation of server clusters in cloud data centers. *IEEE Trans. Serv. Comput.*
- Liu, Z., Chen, Y., Bash, C., Wierman, A., Gmach, D., Wang, Z., Marwah, M., Hyser, C., 2012. Renewable and cooling aware workload management for sustainable data centers. *ACM SIGMETRICS Perform. Eval. Rev.* 40 (1), 175–186 ACM.
- Liu, H., Liu, B., Laurence, T.Y., Man, L., Yuhui, D., Kashif, B., Samee, U.K., 2018. Thermal-aware and DVFS-enabled big data task scheduling for data centers. *IEEE Transact. Big Data* 4 (2), 177–190.
- Longford, G., 2010. Network neutrality vs. network diversity: A survey of the debate, policy landscape and implications for broadband as an essential service for Ontarians. Working paper for the Ministry of Government Services (Ontario). CRACIN nd Web. 27 Jan.
- Möbius, C., Dargie, W., Schill, A., 2014. Power consumption estimation models for processors, virtual machines, and servers. *IEEE Trans. Parallel Distrib. Syst.* 25 (6), 1600–1614.
- Mhedheb, Y., Jrad, F., Tao, J., Zhao, J., Kołodziej, J., Streit, A., 2013. Load and thermal-aware VM scheduling on the cloud. In: *In the Proceeding of International Conference on Algorithms and Architectures for Parallel Processing*. Springer, pp. 101–114 Cham.
- Moore, J.D., Chase, J.S., Ranganathan, P., Sharma, R.K., 2007. Making Scheduling “Cool”: temperature-aware workload placement in data C Zhang S, Chatha K S. approximation algorithm for the temperature aware scheduling problem. *Proc. Int. Conf. Comput.-Aided Des.* 281–288.
- Oxley, M.A., Pasricha, S., Maciejewski, A.A., Siegel, H.J., Burns, P.J., 2016. Online resource management in thermal and energy constrained heterogeneous high performance computing. In: 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). *IEEE*, pp. 604–611.
- Park, K., Pai, V.S., 2006. Comon: a mostly-scalable monitoring system for planetlab. *ACM SIGOPS Operat. Syst. Rev.* 40 (1), 65–74.
- Pierson, J.-M., Stolf, P., Sun, H., Casanova, H., 2019. MILP formulations for spatio-temporal thermal-aware scheduling in Cloud and HPC datacenters. *Cluster Comput.* 1–19.
- Qinghui, T., Gupta, S.K.S., Georgios, V., 2008. Energy-efficient thermal-aware task scheduling for homogeneous highperformance computing data centers: a cyber-physical approach. *IEEE Trans. Parallel Distrib. Syst.* 19 (11), 1458–1472.
- Ranganathan, P., Sharma, R.K., 2005. Making scheduling “cool”: temperature-aware workload placement in data centers.” In *USENIX annual technical conference. General Track* 61–75.
- Rodero, I., Viswanathan, H., Lee, E.K., Gamell, M., Pompili, D., Parashar, M., 2012. Energy-efficient thermal-aware autonomic management of virtualized HPC cloud infrastructure. *J. Grid Comput.* 10 (3), 447–473.
- Saha, S., Majumdar, A., 2017. Data centre temperature monitoring with ESP8266 based Wireless Sensor Network and cloud based dashboard with real time alert system. In: 2017 Devices for Integrated Circuit (DevIC). *IEEE*, pp. 307–310.
- Singh, S., Chana, I., 2016. A survey on resource scheduling in cloud computing: Issues and challenges. *J. Grid Comput.* 14 (2), 217–264.
- Singh, S., Chana, I., Singh, M., Buyya, R., 2016. SOCCER: self-optimization of energy-efficient cloud resources. *Cluster Comput.* 19 (4), 1787–1800.
- Tian, H., Zheng, Y., Wang, W., 2019. Characterizing and synthesizing task dependencies of data-parallel jobs in Alibaba cloud. In: *Proceedings of the ACM Symposium on Cloud Computing*, pp. 139–151.
- Tsai, T.-H., Chen, Y.-S., 2016. Thermal-throttling server: A thermal-aware real-time task scheduling framework for three-dimensional multicore chips. *J. Syst. Softw.* 112, 11–25.
- Wu, G., Xu, Z., 2010. Temperature-aware task scheduling algorithm for soft real-time multi-core systems. *J. Syst. Softw.* 83 (12), 2579–2590.

- Wu, Z., Li, X., Garraghan, P., Jiang, X., Ye, K., Albert, Y.Z., 2016. Virtual machine level temperature profiling and prediction in cloud datacenters. In: 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS). IEEE, pp. 735–736.
- Wu, W., Lin, W., He, L., Wu, G., Hsu, C.-H., 2019. A power consumption model for cloud servers based on elman neural network. *IEEE Transact. Cloud Comput.*
- Wu, H., Zhang, W., Xu, Y., Xiang, H., Huang, T., Ding, H., Zhang, Z., 2019. Aladdin: optimized maximum flow management for shared production clusters. In: In 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, pp. 696–707.
- Xiang, L., Garraghan, P., Jiang, X., Wu, Z., Xu, J., 2018. Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy. *IEEE Trans. Parallel Distrib. Syst.* 29 (6), 1317–1331.
- Xu, M., Buyya, R., 2019. BrownoutCon: a software system based on brownout and containers for energy-efficient cloud computing. *J. Syst. Softw.* 155, 91–103 September.
- Zhao, H., Wang, J., Liu, F., Wang, Q., Zhang, W., Zheng, Q., 2018. Power-aware and performance-guaranteed virtual machine placement in the cloud. *IEEE Trans. Parallel Distrib. Syst.* 29 (6), 1385–1400.
- Zhou, J., Wei, T., 2015. Stochastic thermal-aware real-time task scheduling with considerations of soft errors. *J. Syst. Softw.* 102, 123–133.
- Zhou, J., Cao, K., Cong, P., Wei, T., Chen, M., Zhang, G., Yan, J., Ma, Y., 2017. Reliability and temperature constrained task scheduling for makespan minimization on heterogeneous multi-core platforms. *J. Syst. Softw.* 133, 1–16.
- Zhou, Q., Xu, M., Gill, S.S., Gao, C., Tian, W., Xu, C., Buyya, R., 2020. Energy efficient algorithms based on VM consolidation for cloud computing: comparisons and evaluations. In: Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2020), pp. 1–10 Online Available at: arXiv:2002.04860.



Sukhpal Singh Gill is a Lecturer (Assistant Professor) in Cloud Computing at School of Electronic Engineering and Computer Science (EECS), Queen Mary University of London, UK. Prior to this, Dr. Gill has held positions as a Research Associate at the School of Computing and Communications, Lancaster University, UK and also as a Postdoctoral Research Fellow at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia. Dr. Gill was a research visitor at Monash University, University of Manitoba and Imperial College London. He was a recipient of several awards, including the Distinguished Reviewer Award from Software: Practice and Experience (Wiley), 2018, and served as the PC member for venues such as UCC, SE-CLOUD, ICCCN, ICDICT and SCES. His one review paper has been nominated and selected for the ACM 21st annual Best of Computing Notable Books and Articles as one of the notable items published in computing - 2016. He has published 50+ papers as a leading author in highly ranked journals and conferences with H-index 20. Dr. Gill has reviewed 160+ research articles of high ranked journals and prestigious conferences. His research interests include Cloud Computing, Internet of Things, Energy Efficiency and Machine Learning. For further information on Dr. Gill, please visit: www.ssgill.me.



Shreshth Tuli is an undergraduate student at the Department of Computer Science and Engineering at Indian Institute of Technology - Delhi, India. He is a national level Kishore Vaigyanic Protsahan Yojana (KVPY) scholarship holder for excellence in science and innovation. He has worked as a visiting researcher at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computing and Information Systems, the University of Melbourne, Australia. Most of his projects are focused on developing technologies for future requiring sophisticated hardware-software integration. He has published his research work in top tier journals and conferences, including ACM Transactions on Embedded Com-

puting Systems, Elsevier JSS and FCGS, IEEE CloudCom and Wiley IITL. He is an active reviewer of SPE Wiley, Elsevier JSS and FCGS. His research interests include Internet of Things (IoT), Fog Computing, Network Design, and Artificial Intelligence.



Adel Nadjaran Toosi is a Lecturer in computer systems at Faculty of Information Technology, Monash University, Australia. He received his B.Sc. degree in 2003 and his M.Sc. degree in 2006 both in Computer Science and Software Engineering from Ferdowsi University of Mashhad, Iran and his Ph.D. degree in 2015 from the University of Melbourne. Adel's Ph.D. studies were supported by International Research Scholarship (MIRS) and Melbourne International Fee Remission Scholarship (MIFRS). His Ph.D. thesis was nominated for CORE John Makepeace Bennett Award for the Australasian Distinguished Doctoral Dissertation and John Melvin Memorial Scholarship for the Best Ph.D. thesis in Engineering. His research interests include scheduling and resource provisioning mechanisms for distributed systems.

Currently, he is working on resource management in Software-Defined Networks (SDN)-enabled Cloud Computing.



Felix Cuadrado is a Distinguished Academic (Beatriz Galindo Senior Fellowship) in the Technical University of Madrid (UPM) and a Fellow of the Alan Turing Institute. Prior to joining UPM, he was Senior Lecturer (Associate Professor) in the School of Electronic Engineering and Computer Science at Queen Mary University of London. His research explores the challenges arising from large-scale applications through a combination of software engineering, distributed systems, measurement, and mathematical approaches. He has numerous publications in top tier journals and conferences, including IEEE Transactions on Software Engineering, IEEE Transactions on Cloud Computing, Elsevier JSS and FCGS, IEEE ICDCS, and WWW.

He also has substantial experience in funded research, being Co-PI for European projects (FP7 TRIDEC and H2020 ENDEAVOUR – Internet-scale network management), as well as the EPSRC project EARL (measurements at Internet scale) and Turing Project Raptority.



Peter Garraghan is a Lecturer in the School of Computing & Communications, Lancaster University. His primary research expertise is studying the complexity and emergent behavior of massive-scale distributed systems (Cloud computing, Datacenters, Internet of Things) to propose design new techniques for enhancing system dependability, resource management, and energy-efficiency. Peter has industrial experience building large-scale production distributed systems and has worked and collaborated internationally with the likes of Alibaba Group, Microsoft, STFC, CONACYT, and the UK Datacenter and IoT industry.



Rami Bahsoon is a Senior Lecturer (Associate Professor) at the School of Computer Science, University of Birmingham, UK. Bahsoon's research is in the area of software architecture, cloud and services software engineering, self-aware software architectures, self-adaptive and managed software engineering, economics-driven software engineering and technical debt management in software. He co-edited four books on Software Architecture, including Economics-Driven Software Architecture; Software Architecture for Big Data and the Cloud; Aligning Enterprise, System, and Software Architecture. He was a Visiting Scientist at the Software Engineering Institute (SEI), Carnegie Mellon University, USA (June-August 2018) and was the

2018 Melbourne School of Engineering (MSE) Visiting Fellow of The School of Computing and Information Systems, the University of Melbourne (August to Nov 2018). He holds a Ph.D. in Software Engineering from University College London (2006) and was MBA Fellow in Technology at London Business School (2003–2005). He is a fellow of the Royal Society of Arts and Associate Editor of IEEE Software - Software Economies.



Hanan Lutfiyya is a Professor and the Chair of the Department of Computer Science, University of Western Ontario (UWO), Canada. Her research interests include Internet of Things, software engineering, self-adaptive and self-managing systems, autonomic computing, monitoring and diagnostics, mobile systems, policies, and clouds. She was a recipient of the UWO Faculty Scholar Award in 2006. She is a Past Member of the Natural Science and Engineering Research Council of Canada (NSERC) Discovery Grant Committee, and a Past Member and the Chair of an NSERC Strategic Grants Committee. She was a member of the Computer Science Accreditation Council. She is currently an Associate Editor of the IEEE Transactions on

Network and Service Management and has recently served as the Program Co-Chair for the IEEE/IFIP Network Operations and Management Symposium and the IEEE International Conference on Network and Service Management. She is currently on the steering committee for the Ontario Celebration of Women in Computing Conference.



Rizos Sakellariou obtained his Ph.D. from the University of Manchester in 1997. Since then he held positions with Rice University and the University of Cyprus, while currently he is with the University of Manchester leading a laboratory that carries out research in High-Performance, Parallel and Distributed systems, which over the last 10 years has hosted more than 30 doctoral students, researchers and long-term visitors. Rizos has carried out research on a number of topics related to parallel and distributed computing, with an emphasis on problems stemming from efficient resource utilization and workload allocation and scheduling issues. He has published over 140 research papers, His research has been supported by sev-

eral UK and EU projects and has been on the Program Committee of over 150 conferences and workshops. He values collaboration and a strong work ethic.



Omer Rana is a Professor of Performance Engineering in School of Computer Science & Informatics at Cardiff University and Deputy Director of the Welsh e-Science Centre. He holds a Ph.D. from Imperial College. His research interests extend to three main areas within computer science: problem solving environments, high performance agent systems and novel algorithms for data analysis and management. Moreover, he leads the Complex Systems research group in the School of Computer Science & Informatics and is director of the "Internet of Things" Lab, at Cardiff University.



Shahram Dustdar is Full Professor of Computer Science heading the Research Division of Distributed Systems at the TU Wien, Austria. He holds several honorary positions: University of California (USC) Los Angeles; Monash University in Melbourne, Shanghai University, Macquarie University in Sydney, and University of Groningen (RuG), The Netherlands (2004-2010). From Dec 2016 until Jan 2017 he was a Visiting Professor at the University of Sevilla, Spain and from January until June 2017 he was a Visiting Professor at UC Berkeley, USA. From 1999 - 2007 he worked as the co-founder and chief scientist of Caramba Labs Software AG in Vienna (acquired by Engineering NetWorld AG), a venture capital co-funded software company focused on software for collaborative processes in teams. Caramba Labs was nominated for several (international and national) awards: World Technology Award in the category of Software (2001); Top-Startup companies in Austria (Cap Gemini Ernst & Young) (2002); MERCUR Innovation award of the Austrian Chamber of Commerce (2002). He is founding co-Editor-in-Chief of the new ACM Transactions on Internet of Things (ACM TIoT) as well as Editor-in-Chief of Computing (Springer). He is an Associate Editor of IEEE Transactions on Services Computing, IEEE Transactions on Cloud Computing, ACM Transactions on the Web, and ACM Transactions on Internet Technology, as well as on the editorial board of IEEE Internet Computing and IEEE Computer. Dustdar is recipient of the ACM Distinguished Scientist award (2009), the IBM Faculty Award (2012), an elected member of the Academia Europaea: The Academy of Europe, where he is chairman of the Informatics Section, as well as an IEEE Fellow (2016).



Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He served as a Future Fellow of the Australian Research Council during 2012-2016. He has authored over 625 publications and seven text books including "Mastering Cloud Computing" published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. He also edited several books including "Cloud Computing: Principles and Paradigms" (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index = 134, g-index = 298, 96,000+ citations). "A Scientometric Analysis of Cloud Computing Literature" by German scientists ranked Dr. Buyya as the World's Top-Cited (#1) Author and the World's Most-Productive (#1) Author in Cloud Computing. Dr. Buyya is recognized as a "Web of Science Highly Cited Researcher" for four consecutive years since 2016, a Fellow of IEEE, and Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier and recently (2019) received "Lifetime Achievement Awards" from two Indian universities for his outstanding contributions to Cloud computing and distributed systems. Software technologies for Grid and Cloud computing developed under Dr. Buyya's leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Dr. Buyya has led the establishment and development of key community activities, including serving as foundation Chair of the IEEE Technical Committee on Scalable Computing and five IEEE/ACM conferences. These contributions and international research leadership of Dr. Buyya are recognized through the award of "2009 IEEE Medal for Excellence in Scalable Computing" from the IEEE Computer Society TCSC. Manjrasoft's Aneka Cloud technology developed under his leadership has received "2010 Frost & Sullivan New Product Innovation Award". Recently, Dr. Buyya received "Mahatma Gandhi Award" along with Gold Medals for his outstanding and extraordinary achievements in Information Technology field and services rendered to promote greater friendship and India-International cooperation. He served as the founding Editor-in-Chief of the IEEE Transactions on Cloud Computing. He is currently serving as Editor-in-Chief of Journal of Software: Practice and Experience, which was established 50+ years ago. For further information on Dr. Buyya, please visit his cyberhome: www.buyya.com.