

BTVMP: A Burst-aware and Thermal-efficient Virtual Machine Placement Approach for Cloud Data Centers

Jie Li, Yuhui Deng, Rui Wang, Yi Zhou, Hao Feng, Geyong Min, and Xiao Qin

Abstract—With the rapid growth of cloud computing, frequent workload bursts show an increasing influence on the Quality of Service (QoS) and energy efficiency of cloud-based data centers. Existing virtual machine placement schemes are expected to optimize either QoS or energy efficiency for cloud data centers running under bursty workload conditions. To bridge this gap, we propose a burst-aware and thermal-efficient virtual machine placement technique called *BTVMP*. *BTVMP* adopts a two-step strategy to achieve energy efficiency while assuring QoS. First, *BTVMP* leverages a split-and-recombine algorithm – *SAR* – to deal with bursty workloads. *SAR* prioritizes critical workloads while preventing low-priority workloads from starvation, thereby assuring QoS. Second, *BTVMP* utilizes an enhanced simulated annealing algorithm called *ESA* to offer optimal thermal-efficient virtual machine placement (VMP) solutions, aiming to minimize the energy consumption of data centers. To facilitate estimating energy consumption, we integrate into *BTVMP* a thermal model that takes into account heat re-circulation effects. We conduct extensive experiments with a real-world trace. We compare *BTVMP* with the leading-edge VMP strategies, including Genetic Algorithm (XINT-GA), Power-Aware and Performance-Guaranteed Virtual Machine Placement (PPVMP), Peak Load Scheduling Control Method (PLSC), First Come First Serve (FCFS), and GReedy based scheduling Algorithm miNimizing Total Energy (GRANITE). The experimental results unveil that *BTVMP* not only enhances QoS but also exhibits superb energy efficiency. In particular, *BTVMP* reduces PLSC's workload delay and FCFS's critical workload delay by 18% and 11%, respectively. Moreover, *BTVMP* lowers the total energy consumption of the three alternative algorithms –GRANITE, XINTGA, PPVMP, and PLSC – by anywhere between 27.8% and 49.4%.

Index Terms—QoS guarantee, Virtual Machine Placement, Bursts, Tail Latency, Cloud Data Center.

I. INTRODUCTION

J. Li is with the Department of Computer Science, Jinan University, Guangzhou, 510632, China, and is with School of Software, South China Normal University, Foshan, Guangdong, 588225, China, E-mail:lijiegxm11@163.com

Y. Deng is with Department of Computer Science, Jinan University, Guangzhou 510632, China, and is also with Wuhan National Laboratory for Optoelectronics, Wuhan, China. E-mail:tyhdeng@jnu.edu.cn (Corresponding author)

R. Wang, and H. Feng are with the Department of Computer Science, Jinan University, Guangzhou, China, 510632. E-mail: dcfh_wr@163.com, fh76020634@163.com

Y. Zhou is with the TSYS School of Computer Science, Columbus State University, Columbus, GA, U.S. E-mail: zhou_yi@columbusstate.edu

G. Min is with the Department of Computer Science, College of Engineering, Mathematics, and Physical Sciences, University of Exeter, Exeter, EX44QF, U.K. E-mail: g.min@exeter.ac.uk

X. Qin is with the Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849-5347 US. E-mail: xqin@auburn.edu

CLOUD computing has been widely used in various fields including medical treatment, bio information, communication, scientific computing, internet-of-things technology, IoT, and to name just a few [1] [2]. Furthermore, the increasing use of computing-intensive applications generates colossal amounts of workloads, which suddenly exceed the maximum workload that current servers can handle: the server systems are forced into an overload state. Because expanding server resources requires a certain time period, these overloaded servers are unable to swiftly process data: the problem of service load imbalance occurs once the workload in a cloud data center increases suddenly [3].

The current method of coping with bursty workloads is to migrate workload from overloaded servers to underloaded servers [4]. Although this method enables in-situ processing of delay-sensitive workloads, which also causes extra energy if the workload is migrated among servers. The long-tail effect – the starvation problem – of low-priority workloads has become a dominating issue affecting the QoS of cloud datacenters when workload bursts emerge. For example, Wang *et al.* [5] illustrated that the long-tail effect affects computational visualization systems that are already prevalent in the real world and becomes a fatal problem for driving vision systems applied to complex scenes. The long-tail effect is a common nuisance for users, as articulated by Silva *et al.* [6]. Operators might be able to accomplish this by rearranging tasks that take longer to complete. Cloud providers ought to pledge sufficient resources to meet clients' performance demands once clients and service providers sign a Service-Level Agreement (SLA) on the availability and performance of hosted applications [7]. A common scheduling strategy adopted by cloud service providers to cope with bursty workloads is to preferentially allocate limited computing resources to high-priority workloads. This simple priority-based scheduling strategy, however, tends to provoke starvation of low-priority workloads that suffer from unacceptable delay.

Operations under bursty workload conditions to render available on-demand cloud services has become a challenging scenario for cloud data centers [8]. When workload bursts emerge, cloud service providers usually adopt two strategies: (i) activate more computing nodes (i.e., server) to accommodate bursty execution requests or (ii) delay a portion of workload that exceeds a data center's processing threshold until computing resources become available. Unfortunately, both strategies bring forth adverse impacts. The former solution must take a time period to revamp the number of computing

nodes. During this period, the data center is constantly running under overload conditions. Such an overload condition enlarges workload delay, thereby jeopardizing QoS to clients. On the other hand, workload-delay algorithms often lead to greater energy consumption due to extended resource wait times [9] [10] [3]. These observations motivate us to pay heed to both the QoS and energy efficiency of cloud data centers when making virtual machine placement (VMP) decisions.

We address the aforementioned issue by proposing a novel split-and-recombine algorithm or *SAR*. Unlike the simple priority-based categorizing strategy, we categorize incoming workloads by workload execution time and workload importance. Moreover, we monitor and dynamically adjust workload priorities. In doing so, we successfully eliminate the long tail effect while guaranteeing the QoS for a majority of workloads. Low-priority workloads, such as simple web content serving and batch workloads, are defined as ordinary workloads characterized by delay tolerance. Correspondingly, workloads with higher priorities are defined as key workloads, which have delay-sensitive characteristics. Example key workloads include high numeric intensive workloads, analytical workloads, database workloads, and instant messaging [11]. The purpose of dividing workloads into the key and ordinary families is to mitigate performance degradation caused by overload by using two layers of request queuing [3]. The QoS of critical workloads can be guaranteed by scheduling the loads as a matter of priority in one queue, and the ordinary workloads with the long-tail effect in another queue are prone to promote priorities. Identifying whether an incoming workload is a key workload or an ordinary workload is extremely paramount. Fortunately, according to the prompts of clients or their applications, or through the access history information of data mining techniques, the incoming workload classification problem can be ironed out [12] [13].

There is a growing awareness of the necessity to conserve the energy consumption of cloud data centers [14] [15]. An increasing number of VMP are proposed to minimize the energy consumption of cloud data centers [16]. Most energy-saving VMP technologies share a common strategy by optimizing the utilization of computing resources – bandwidth resources, CPU usage, network resources, memory usage, and the like [17]. Related research states [18] [19] that the total energy consumption of a data center mainly comes from IT equipment energy and cooling system energy, of which IT equipment energy accounts for about 40% of the total energy, and cooling system energy accounts for about 50% of the energy of the data center. IT equipment energy encompasses mainly server energy and network hardware energy, of which network hardware energy accounts for about 10% of the energy of the data center. Unfortunately, the cooling energy consumption of cloud data centers is rarely of concern to these energy-saving VMP technologies. Underutilized computing resources also play a vital role in the overall energy consumption, especially for cloud data centers [20]. Furthermore, cloud service providers furnish VM resources to users, whose requirements are ensured by the assigned VMs. Multiple VMs sharing the same server resources can help improve resource usage and reduce energy consumption, thereby leading

to resource competition and performance degradation [21]. Therefore, VMP has a vital impact on performance, resource utilization, and power consumption of cloud data centers. Ensuring QoS and energy efficiency becomes increasingly indispensable for building cloud data centers running under bursty workload conditions.

In this study, we delve into a thermal-aware heuristic algorithm to attain the optimal energy-efficient VMP toward the goal of ensuring QoS and bolstering the energy efficiency of cloud data centers. The main contributions of this study are summarized below:

- We propose a burst-aware and thermal-efficient virtual machine placement approach - *BTVM*P. *BTVM*P optimizes QoS performance while minimizing energy consumption of cloud data centers running under bursty workload conditions.
- We adopt a two-step strategy to implement our burst-aware and thermal-efficient virtual machine placement approach for cloud data centers. In the first step, algorithm *SAR* splits and recombines workloads to prioritize critical workloads while avoiding low-priority workloads from starvation. In the second step, the enhanced simulated-annealing-based heuristic algorithm or *ESA* to reach an optimal virtual machine placement solution. *ESA* minimizes the total power consumption by taking heat re-circulation effects into account.
- We evaluate *BTVM*P in terms of workload delay, energy efficiency, and execution time. The experimental results reveal that *BTVM*P remarkably reduces workload delay of FCFS and PLSC by an average of 81% and 82%, respectively. Moreover, *BTVM*P minimizes the total power consumption by 27.8%-49.4% compared with the state-of-art alternatives XINT-GA, PPVMP, PLSC, and GRANITE.

The rest of this paper is organized as follows. Section II summarizes related work. Section III introduces the system overview, where the *BTVM*P strategy architecture, split-and-recombine method, and thermal and energy-cost model are presented. Section IV presents our two-step approach *BTVM*P for dealing with load bursts and minimizing energy consumption of data centers. Performance evaluation is discussed in Section V. Finally, we conclude, in Section VI, this paper along with our future research directions.

II. RELATED WORK

Essential factors burst-awareness and thermal efficiency becomes increasingly pivotal to manage and operate cloud data centers of guaranteed QoS. Though various virtual machine workload placement approaches have been proposed to cope with bursty workloads or achieve high energy efficiency, few techniques pay attention to these two indispensable factors at the same time to offer guaranteed QoS and energy-efficient cloud services. In this section, we pinpoint the primary design principles as well as the performance issues. Table I summarize the major technological discrepancies between *BTVM*P and the existing strategies reported in the literature. We qualitatively compare *BTVM*P against the alternative

strategies from the perspectives of burst-awareness, VMP, thermal awareness, starvation avoidance, and energy efficiency, respectively.

A. Bursty Workloads and QoS Assurance

Much attention has been paid to coping with bursty workloads and OoS improvement in the realm of virtual machine placement in cloud data centers. For instance, Rahmani *et al.* [22] proposed a burst-aware VMP method, which strives to mitigate the negative impact of workload bursts. Luo *et al.* [23] investigated a VM consolidation technique catering for workload bursts. However, these two schemes fall short of ensuring QoS for users of cloud services. Voorsluys *et al.* [29] designed a VMP scheme with a purpose of advancing manageability and fault tolerance. Unfortunately, Voorsluys achieves his stated goals at the cost of high CPU utilization of hosting physical machines (PMs), which leads to substantial performance degradation. In addition to the above-mentioned schemes, Lu *et al.* [8] built a decomposition algorithm and a recombination algorithm to handle bursty loads for ensuring the QoS of cloud data centers. Zhu *et al.* [10] introduced a system named WorkloadCompactor to consolidate multiple workloads onto a storage server while meeting tail latency performance requirements. Jang *et al.* [9] presented a Silo system to impose low overhead by governing a ratio between hosts and tenants. Hu *et al.* [3] developed PLSC that splits and recombines incoming workloads based on workload criticality. PLSC, designed to guarantee QoS for critical workloads, fails to take care of low-priority workloads. Beloglazov *et al.* [30] presented an optimal offline algorithm to maximize average workload migration time under a certain QoS constraint. Deng *et al.* [4] dealt with the bursty workload problem via a two-stage strategy, in which workload bursts are first migrated to low-utilization servers, then all workloads will be processed in parallel to reshape efficiency. Unlike our BTVMP, the aforementioned strategies merely focus on bursty workload to improve QoS, ignoring the heat recirculation effect in cloud data centers.

B. VMP for Improving Energy Efficiency of Data Centers

An increasing number of researchers embarked on enhancing the energy efficiency of data centers through virtual machine placement [31]. Azizi *et al.* [32] introduced a heuristic algorithm to optimize the placement of virtual machines into physical machines to minimize the total energy consumption of the data center by reducing the number of physical machines. Zhao *et al.* [21] presented a power-aware and performance-assured VM placement strategy to boost the system performance of VMs and minimize the power consumption of physical machines. Xu *et al.* [28] presented a self-adaptive approach for the resource management of interactive workloads and batch workloads. This work aims to minimize the carbon footprint of data centers by cutting down on brown energy utilization.

The above methods ignore the burst workload and the effect of heat recirculation, which can significantly incur an increase in the energy consumption of data centers.

C. Thermal-aware VM Placement or Task allocation in Data Centers

Thermal-aware management technology is essential in data centers. For example, Guo *et al.* [33] proposed a temperature-aware VDC embedding scheme to mitigate local hot spots. The scheme aims to minimize both the maximum temperature of hot air drawn into each rack and the total power consumption of IT equipment. Ilager *et al.* [34] introduced the Energy and Thermal-Aware Scheduling (ETAS) algorithm, which is based on the GRASP meta-heuristic. It dynamically consolidates virtual machines to minimize overall energy consumption while proactively preventing hotspots. In [35], Chen *et al.* considered the Power and thermal-aware virtual machine scheduling optimization problem and proposed a VM scheduling scheme PTDS. This scheme depends on the real-time detection of computing node temperature and resource utilization under the heat recirculation effect to dynamically adjust the VM consolidation. Feng *et al.* [26] built a simulated annealing-based algorithm, intending to obtain optimal thermal-efficient VM placement for data centers. A thermal-aware workload scheduling mechanism was implemented by Tang *et al.* to reduce cooling costs by placing multiple tasks across different computing nodes [36]. Li *et al.* [37] constructed a thermal-aware computing node failure model and offered a hybrid genetic algorithm and an intelligent optimization algorithm for simulated degradation to achieve an efficient workload distribution while minimizing the total energy consumption of the data center.

While the above approach considers the problem of constructing thermal-aware models in cloud data centers, it does not consider the problem of how bursty loads are handled. For this regard, we propose the VM placement strategy for burst aware and thermal-efficient, which first builds a thermal awareness model for cloud data centers, then schedules the processed burst workloads into VMs, and finally leverages an improved simulated annealing algorithm to reasonably place VMs into servers so as to achieve the optimized balance between the overall energy efficiency and QoS for the data center.

III. SYSTEM OVERVIEW

Workloads are marked as workload bursts if the amount of the workloads imposed on a data center exceeds the processing capacity of the data center. The overarching goal of BTVMP is to keep the number of bursty workloads under the processing capacity of the data center while ensuring QoS and high energy efficiency of cloud data centers. On the one hand, to keep the number of bursty workloads under the processing capacity, BTVMP prioritizes critical workloads while postponing low-priority workloads that are prone to long-tail effects. BTVMP monitors and dynamically promotes the priority of non-critical workloads to prevent non-critical workloads from starvation. Moreover, BTVMP classifies and recombines critical and non-critical workloads to meet QoS requirements for delay-sensitive workloads. On the other hand, to minimize the energy consumption of cloud data centers, BTVMP considers heat recirculation effects amid the procedure of making virtual machine placement decisions. To facilitate thermal-efficient VMP

TABLE I: Comparisons of BTVMP and the existing works

Techniques	Burst-awareness	VMP	Thermal awareness	Starvation avoidance	Energy efficiency
Burstiness-VMP [22]	✓	✓	✗	✗	✓
FFD-Markov [23]	✓	✓	✗	✗	✓
RTT-MS [8]	✓	✓	✗	✓	✗
WC-System [10]	✓	✗	✗	✓	✗
Silo-System [9]	✓	✓	✗	✓	✗
PLSC-QoS [3]	✓	✓	✗	✗	✓
SG-VMC [24]	✗	✓	✗	✓	✓
PPVMP [21]	✗	✓	✗	✓	✓
MA-VMP [25]	✗	✓	✗	✗	✓
GRANITE [18]	✗	✗	✓	✗	✓
SAG-VMP [26]	✗	✓	✓	✗	✓
OEMACS-VMP [27]	✗	✓	✗	✗	✓
Burst-DEN [4]	✓	✗	✗	✗	✗
Brownout-Deferring [28]	✓	✓	✗	✗	✓
BTVMP	✓	✓	✓	✓	✓

decision-making, we propose an efficient heuristic algorithm to solve the VMP optimization problem.

A. BTVMP Strategy Architecture

Aiming to strengthen QoS while enhancing energy efficiency, we design the architecture of BTVMP – illustrated in Fig. 1 – to deal with workloads through two key steps: (1) workload splitting and recombining, and (2) virtual machine placement. The design of BTVMP is driven by two principles. First, BTVMP splits and recombines workload queues by SAR algorithm to prioritize critical workloads while preventing low-priority workloads from starvation. It is noteworthy that the critical workload and ordinary workloads are determined based on the Service Level Agreement (SLA) requirements of the workload. If the SLA requirements of the workload have higher requirements for a workload, then it can be defined as a critical workload or key workload. Second, an enhanced heuristic algorithm ESA is utilized by BTVMP to place VMs in an energy-efficient manner.

Step 1: Splitting and Recombining. To wrestle with bursty workloads, in this step, BTVMP leverages SAR to classify workloads based on both workload priority and workload execution time (See also Fig.2). More specifically, SAR categorizes incoming workloads into two camps, namely, *ordinary workloads* and *key workloads*. Then, based on workload execution time, these ordinary workloads and key workloads are classified into three types: *long-execution-time workloads* (i.e., *Key load 1 and Ordinary load 1*), *medium-execution-time workloads* (i.e., *Key load 2 and Ordinary load 2*), and *short-execution-time workloads* (i.e., *Key load 3 and Ordinary load 3*). Next, SAR prioritizes key workloads while monitoring the execution rate of ordinary workloads to avoid ordinary workloads from starvation. The priority of ordinary workloads will be increased over time such that low-priority workloads gain more chances to be scheduled for execution. Last, key and ordinary workloads of various execution-time types are recombined appropriately into VMs for future scheduling. For example, a low-priority short-execution-time ordinary workload is combined with a long-execution-time key workload into a VM for future scheduling and execution. It is noteworthy

that with the purpose of ensuring QoS, a fixed proportion of computing resources are allocated to execute ordinary workloads, the starvation problem of which is avoided. Besides, because short-term workloads are sensitive to delay, short-term workloads take precedence over other ordinary workloads to attain VM resources for execution.

Step 2: Thermal-efficient Virtual Machine Placement.

The recombined workload queues generated from the splitting-and-recombining step are encapsulated in VMs for further placement on computing servers. In this step, BTVMP makes use of our proposed ESA algorithm to assign VMs on physical servers in an energy-efficient way.

B. Split-and-recombine Method

The workload reshaping process encompasses the split and recombine phases, as shown in Fig. 2. Our SAR algorithm processes the workload process into two classes in the split phase, identified by the queues Q1 and Q2, respectively. Requests belonging to different classes are directed to different queues. Ordinary workloads with long execution times belonging to Q1 requests require elevated priority to obtain response time or latency guarantees, while key workloads belonging to Q2 requests will be served in a best-effort manner. In the recombination phase, the workloads of the two classes of requests are multiplexed optimally to meet their respective performance constraints. A large number of data centers adopt the FCFS (first-come, first-served) workload scheduling strategy [38]. Using FCFS, however, leads to a situation in which key workloads cannot be processed with guaranteed QoS requirements under burst workload conditions.

Unlike FCFS, our SAR-based BTVMP strategy is designed to warrant the QoS of key workloads regardless of workload conditions (see Fig. 2). In particular, SAR first classifies incoming workloads into two types in light of workload importance. Then SAR categorizes the workloads into three types according to workload processing time, forming six types of workloads. Such two judicious classifications enable BTVMP to keep high QoS even under bursty conditions. More prosaically, when load burst arrives, SAR first processes key workloads and ordinary workloads; then, SAR diverts attention

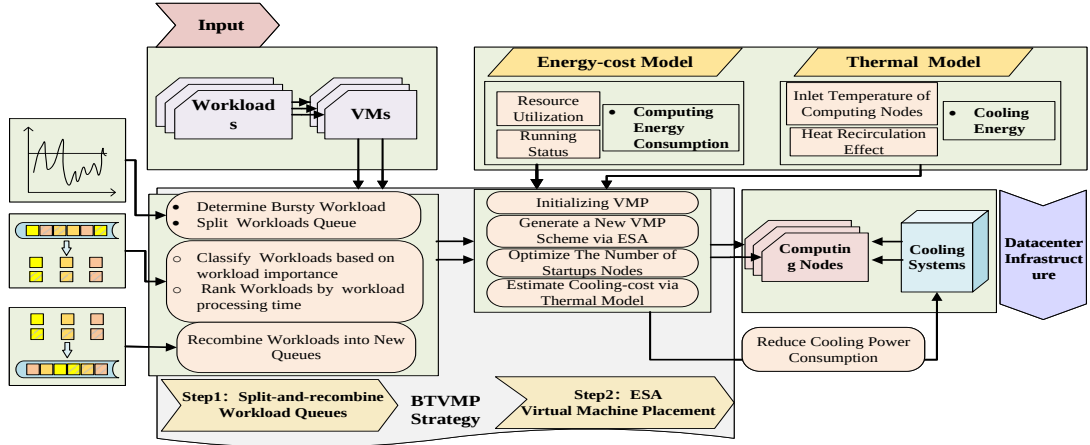


Fig. 1: BTVMP strategic architecture diagram

to ordinary workloads to avert the excessive delay of ordinary workloads. Moreover, SAR determines the execution order of workloads by monitoring and dynamically adjusting the ordinary workload priorities. The priority of a workload is formulated as:

$$Z = \frac{w + \sum T}{w} \quad (1)$$

where Z denotes the priority of the workload to be executed and a high Z value indicates high priority; $\sum T$ represents the total delay time of the queued workload; w denotes the expected completion time for a queued workload, and its computation is derived from the individual workload dwell time [3].

C. Thermal and Energy-cost Models

To minimize the total energy consumption yielded by VMP, we delve into an energy-cost model. With the purpose of accurately estimating the total energy cost of various VM placements, we incorporate the heat recirculation effect coupled with VM placements into the thermal model [18] [26] [36]. We denote heat re-circulation as a cross-interference matrix $A = [\alpha_{ij}]_{n \times m}$ shown in Fig. 3. Please note that we use terms *node* and *server* interchangeably throughout this paper. Matrix A can be calculated as

$$K \cdot (\vec{T}_{out} - \vec{T}_{sup}) = A^T \cdot K \cdot (\vec{T}_{out} - \vec{T}_{sup}) + \vec{P}_i \quad (2)$$

where A^T denotes the transpose matrix of A ; $\vec{T}_{out}$ is the outlet temperature matrix of data center nodes, expressed as $\vec{T}_{out} = [T_{out}^1, T_{out}^2, \dots, T_{out}^n]$. $\vec{T}_{sup}$ represents the supply temperature of the cooling system, formulated as $\vec{T}_{sup} = [T_{sup}, T_{sup}, \dots, T_{sup}]$; $\vec{P}_i$ is the energy required by node i , denoted as $\vec{P} = [P_1, P_2, \dots, P_n]^T$. In addition, we take into account of the thermodynamic constants expressed as a diagonal matrix $K_{n \times n} = \text{diag}(K_1, K_2, \dots, K_n)$, where $K_i = \rho f_i C_p$, ρ is air density (g/m^3), f is airflow rate (m^3/g), and C_p is air heat ($Jg^{-1}K^{-1}$).

After the splitting and recombining stage is accomplished, workloads are energy-efficiently mapped to VMs for placement on multiple computing nodes in a data center. To simplify

the energy-cost optimization of VM placement, we assume, without losing generality, that all VMs are homogeneous. Given the number of workloads, we denote the total number of the generated VMs in a time period as C_{Total} . For computing nodes powered by multicore processors, we simply divide the total number of VMs by the number of cores per node. Thus, our ESA places the workloads on n nodes, each of which encompasses up to m VMs, each of which consists of c_i -sized workloads. Scheduling results are expected to meet the following constraints:

$$\sum_{i=1}^n c_i - C_{Total} = 0, \text{ and } c_i \leq m \quad (3)$$

where c_i represents a subtask of VMs machines executed by i -th computing node. The heat re-circulation effect leads to an undesired inlet temperature increase. Thus, we formulate the relationship among the inlet temperature of computing nodes, the air supply temperature, and the VMP as:

$$\max \vec{T}_{sup}, \vec{T}_{sup}(c) = T_{in} - Db - D\vec{c}a \quad (4)$$

Because we derive the heat distribution matrix from Equation 2 as $D = [(K - A^T K)^{-1} - K^{-1}]$ and obtain $\vec{T}_{in}$, Db , and a from the data center, the ESA algorithm is capable of calculating the optimal VMP – vector $\vec{c} = \{c_1, c_2, \dots, c_n\}$ – to maximize the air supply temperature of a cooling system while keeping inlet temperatures of computing nodes below a stipulated redline temperature. These salient features enable BTVMP to remarkably conserve cooling energy consumption because air supply temperature plays a pivotal role in cooling cost and energy savings in data centers: every 1 °C increase in supply temperature can save up to 4.3% - 9.8% of cooling cost [39].

The total power consumption of a data center contains two parts, namely, the power consumed by its computing nodes and the power consumed by the cooling system [36]. Power consumed by each computing node can be expressed as $\vec{P}_i = b + c_i a$. Assuming the data center contains n server, we

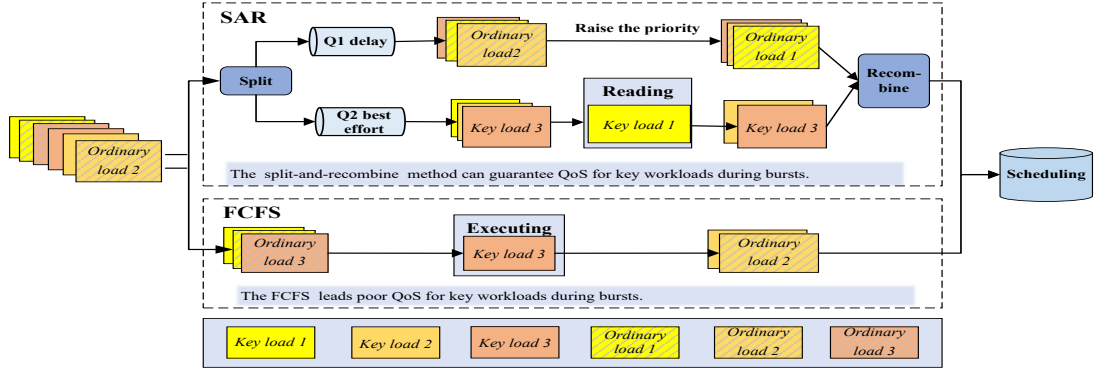


Fig. 2: Schematic diagram of the split-and-recombine method

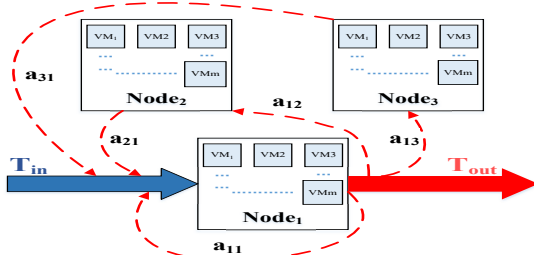


Fig. 3: Heat-recirculation between neighbouring nodes

measure the total power consumption caused by the computing nodes as

$$P_{IT} = \sum_{i=1}^n \vec{P}_i \quad (5)$$

The power consumption of the cooling system is modeled in Equation 6 as a function of computing power consumption and the coefficient (see more details in [40]).

$$P_{Cooling} = \frac{P_{IT}}{CoP(T_{sup})} \quad (6)$$

where coefficient $CoP(T_{sup})$ is a function of the air supply temperature T_{sup} :

$$CoP(T_{sup}) = 0.0068 \cdot T_{sup}^2 + 0.0008 \cdot T_{sup} + 0.458 \quad (7)$$

The overarching goal of our proposed BTVM is to minimize the total power consumption P_{Total} by maximizing air supply temperature $\vec{T}_{sup}$ while keeping inlet temperature $\vec{T}_{in}$ below a certain redline temperature. Not surprisingly, virtual machine placement is a critical factor in achieving such an energy efficiency goal. With Equations 2 - 7 in place, the objective function of BTVM is expressed as:

$$\begin{aligned} P_{Total}, P_{Total} &= P_{IT} + P_{Cooling} \\ \text{Minimize} &= \left(1 + \frac{1}{CoP(T_{sup})}\right) \sum_{i=1}^n \vec{P}_i \end{aligned} \quad (8)$$

D. Service Level Agreement Violation (SLAV) Model

When the VMs are on servers within a data center, it is important to consider Service Level Agreement Violation (SLAV). SLAV defines the expectations and commitments

regarding the QoS that the data center or cloud provider must deliver to its customers. These agreements typically specify parameters such as availability, performance, response times, and reliability. To ensure that SLAs are met, data center operators and administrators must consider that the physical server selected to host a VM has sufficient resources (e.g., CPU, memory, storage, and network bandwidth) to meet the performance requirements outlined in the SLAV. Overloading of VMs in the server will cause migration of VMs, which will lead to SLA violation issues. For this reason, we construct the SLAV model to characterize the performance degradation of the server due to the overloading. problem [35] [41].

$$SLAV(t) = SLATAS(t) * PDM(t)$$

$$SLATAS = \frac{1}{N} \sum_{i=1}^N \frac{T_{s_i}}{T_{a_i}} \quad (9)$$

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}}$$

where $SLAV(t)$ represents the overall SLA violation at time t in a cloud data center. $SLATAS(t)$ denotes the SLA violation time for each active server at time t . $PDM(t)$ expresses the performance degradation due to VM migrations at time t . N represents the number of servers in the data center. T_{s_i} indicates the SLA violation caused by server i having a CPU utilization rate of 100% throughout the time period. T_{a_i} represents the CPU capacity of the active server, indicating the workload processing status during the entire time period. M represents the number of virtual machines (VMs) on server i . C_{d_j} : Denotes the capacity of CPU performance degradation caused by VM migration. C_{r_j} represents the total CPU capacity required by VM requests.

Excessive consolidation of VMs into servers leads to a degradation in the QoS and potentially results in SLA violations. To mitigate these problems, the available resources of each server, such as processing power, need to be carefully determined. To maintain a safe zone for VMP, where service quality degradation is prevented, each server's local manager calculates a safe zone criterion for any deployed VM. This criterion is determined using interval estimation and confidence levels, which help make decisions about predicting the occurrence of overhead on the server. The specific formula

for calculating the safe zone of a VM $V_{vm\text{sz}}$ is expressed as follows [42]:

$$RPP_i = S_{am} - \sum_{\substack{j=1 \\ j \neq i}}^N (V_{pm}^j - V_{om}^j) \quad (10)$$

$$V_{vm\text{sz}}^i = RPP_i - (V_{pm}^i - V_{am}^i)$$

where S_{am} , V_{pm} , V_{om} denote the server's available processing power, the predicted processing power by each VM, the allocated processing power to each VM, respectively. i, j, N represents the number of servers index, the number of VMs index, and the maximum number of VMs for a single server. RPP_i indicates the remaining processing power for VM^i . If $V_{vm\text{sz}} \geq 0$ means the minimum acceptable value for a VM to be placed in the safe zone. The higher the value of $V_{vm\text{sz}}$, the lower the probability of QoS degradation and vice versa.

IV. ALGORITHM DESIGN OF BTVMP

To enhance QoS while minimizing the power consumption of cloud data centers under bursty workloads, BTVMP adopts a two-step strategy to explore an optimal VMP solution. BTVMP first uses the algorithm SAR to deal with bursty workloads with the objective of curbing workload delay. Then, BTVMP employs the algorithm ESA to obtain optimal VMP solutions that minimize the overall energy consumption of data centers.

A. The Split-and-Recombine Algorithm – (SAR)

The primary function of the proposed SAR is to safeguard the QoS of cloud data centers in case of bursty workloads. SAR outlined in Algorithm 1 first categorizes workloads arriving at a data center into key workloads and ordinary workloads according to workload importance (see Lines 1 - 3 in Algorithm 1). In the algorithm, $DC\text{value}$ indicates the threshold value set for the maximum workload that can be processed by the computing nodes of a data center, and $scale_v$ represents the maximum percentage of reserved resources that key/ordinary workloads are handled by computing nodes of the data center. Then, key workloads are prioritized thanks to their important nature. If a key workload's computing resource demand can be satisfied by available computing resources of the cloud data center, the workload is put into the processing queue. Otherwise, the workload is marked as a key workload with excessive cloud computing resource requests (see Lines 4 - 5 in Algorithm 1). Next, SAR invokes Algorithm 2 to sort and schedule ordinary workloads and key workloads with excessive cloud computing resource requests in accordance with workload processing time (see Lines 6 - 7 in Algorithm 1). Last, SAR calculates and updates the processing delay of workloads to dynamically adjust the priorities of ordinary workloads, the starvation of which is fended off (see Lines 8 - 10 in Algorithm 1). The time complexity of SAR – Algorithm 1 – is $O(n)$.

Algorithm DealWorkload depicted in Algorithm 2 is in charge of sorting and scheduling remaining workloads – ordinary workloads and key workloads with excessive requests of cloud computing resources. The remaining workloads are

Algorithm 1 : Split-and-recombine algorithm (SAR)

Input: Incoming workloads request queue M

Output: Workloads queue after split K . Workloads queue with maximum threshold range after recombining N ;

- 1: **if** ($len(M) > DC\text{value}$) **then** //Determine whether it is a burst load;
- 2: $KeyLoad = Sort.getKeyWorkload(M)$;
- 3: $OrdLoad = Sort.getOrdinaryWorkload(M)$; // Read incoming workloads and assign loads to workloads queues of the key or the ordinary workloads queues.
- 4: **if** ($len(KeyLoad) \leq DC\text{value} * scale_v$) **then** // Determine whether the key workload exceeds reserved processing resources;
- 5: $N.append(KeyLoad)$; // Add key workloads to the processing queue.
- 6: $DealWorkload(OrdLoad, N, K)$; // Sort the ordinary workloads into time cost categories;
- 7: $DealWorkload(KeyLoad, N, K)$; // Sort the key workloads into time cost categories;
- 8: **end if**
- 9: **end if**
- 10: $RecordDelayTime()$; // Record delay time;
- 11: $RenewNextDC\text{value}()$; // Calculate the number of workloads that take more than I unit of time and update the processing threshold at the next moment;
- 12: **return** N, K

first classified by DealWorkload according to processing time (see Line 1 in Algorithm 2). The function $TimeSplit()$ divides key workloads from ordinary workloads in light of unequal workload processing time. Then, for a light workload with a short processing time, DealWorkload checks whether the workload's resource request tops the resource threshold of the data center. If the answer is no, DealWorkload places the workload in the processing queue. Otherwise, the workload is appended to the waiting queue (see Lines 2 and 3 in Algorithm 2). Workloads requiring medium-timescale processing time are managed in a similar way (see Lines 4 - 13 in Algorithm 2). The function $Renew()$ means to update the value of the queue. It is straightforward to infer the DealWorkload's time complexity, which is $O(n)$.

B. The ESA Algorithm

VMP has immense impacts on the energy efficiency of cloud data centers. To conserve the energy consumption of data centers, we put forth an enhanced heuristic algorithm coined as *ESA* to derive an optimal virtual machine placement decision. We design *ESA* anchored on the *Simulated Annealing* algorithm (SA) – a powerful means to find a global optimum among a significant number of local optima [43]. The rationale behind the design of *ESA* based on SA is two-fold. First, SA avoids falling into local optima by allowing momentary performance degradation in its iterative search for global optima. Second, SA has been proven as an effective approach to obtaining an energy-efficient VMP solution for data centers [44] [45].

Algorithm 2 : DealWorkload

Input: *OrdLoad, KeyLoad*,

Output: Workloads queue after split *K*. Workload queue with maximum threshold range after recombining *N*;

```

1: [OTk1, OTk2, OTk3] = TimeSplit(OrdLoad, KeyLoad); // Divide ordinary
   workloads and key workload queues into three categories
   on the basis of time-consuming;
2: Function TimeSplit();
3: if (len(OTk1) <= Re(DCvalve)) then // If the work-
   load that takes 1 unit of time does not exceed the current
   remaining processing threshold of data centers;
4:   Renew(N); // Add the queue to the processing queue
   N, and update N;
5:   if (len(OTk2) <= Re(DCvalve)) then // If the
   workload that takes 2 unit of time does not exceed the
   current remaining processing threshold of data centers;
6:     Renew(N);
7:     Renew(K); // Add the part that exceeds the pro-
   cessing threshold to the waiting queue K, and update K
   ;
8:   if (len(OTk3) <= Re(DCvalve)) then // If the
   workload that takes 3 unit of time does not exceed the
   current remaining processing threshold of data centers;
9:     Renew(N);
10:    Renew(K); // Add the part that exceeds the
   processing threshold to the waiting queue K, and update
   K ;
11:   end if
12: end if
13: else
14:   Renew(N);
15:   Renew(K);
16: end if

```

While the SA algorithm is widely deployed to search for global optimal solutions, it exhibits two primary shortcomings. SA starts its optimum searching with a single random solution, which is likely to significantly affect search performance. Moreover, using a single solution search strategy to explore solution space leads to an intolerably long search time [46]. To address these two issues, we developed ESA as an extension of SA. Importantly, EAS has the following three distinctive perks and attributes.

- 1) To speed up the convergence of virtual machine placement optimization, ESA initializes an initial solution with a relatively large value – a position near the top of the rack.
- 2) To quest for an optimal VMP solution, during ESA's iterative exploration in the search space, ESA changes the direction of its new solution towards other directions of the current solution.
- 3) To enhance the utilization of computing nodes, ESA judiciously assigns VMs on a set of active nodes while powering off low-utilization nodes.

The ESA algorithm takes the following four steps to energy

efficiently distribute VMs on computing nodes of cloud data centers.

Step1: Setting Control Parameters. The parameters, including initial temperature T_0 , cooling rate q , and equilibrium state temperature T_{end} , are initialized. It is noteworthy that T_0 and T_{end} are two parameters controlling the iteration of the ESA algorithm.

Step2: Generating Initial Solution. A random initial workload distribution $\vec{c}_1$ is generated. And $\vec{c}_1$ must be subject to constrain $\sum_{i=1}^n c_i = C_{Total}$, where C_{Total} denotes the total number of generated VMs in a time period (see also Lines 1 - 3 in Algorithm 3).

Step3: Generating New Solutions. In each iteration, ESA produces a new solution s_2 , followed by utilizing the *Metropolis* criterion to determine whether or not to accept the new solution s_2 . The Metropolis criterion is expressed as:

$$P = \begin{cases} 1 & \Delta df < 0 \\ \exp(-\frac{\Delta df}{T}) & \Delta df \geq 0 \end{cases} \quad (11)$$

where $\Delta df = f(s_2) - f(s_1)$; $f(s)$ is the fitness function; s_1 and s_2 denote the old solution and the new solution, respectively. If $\Delta df < 0$, ESA accepts the new solution with probability 1. Otherwise, ESA accepts the new solution with a probability of $\exp(-\frac{\Delta df}{T})$ (see Lines 5 - 13 in Algorithm 3). Please note that since the objective function of SA is expressed as $f(s) = T_{in} - Db - D\vec{c}a$, the fitness function of the initial solution can be written as $f(s_1) = T_{in} - Db - D\vec{c}_1a$.

Step 4: Temperature Reduction and Output Final VMP. ESA lowers the temperature according to cooling rate q . Thus, we have $T_{new} = q \cdot T_{old}$. If T is below end temperature T_{end} , ESA stops the iteration and outputs the current VMP solution. Otherwise, the iteration continues as outlined in Lines 4 and 14 - 16 in Algorithm 3.

TABLE II: Notation for algorithm ESA

Notation	Description
T_0	Initial temperature
T_{end}	Final temperature
T	Solid temperature
t	Counter
q	Cooling rate
$Iter$	Iteration times
S	Solution vector of VMP to server

V. EXPERIMENTAL RESULTS

In this section, we evaluate the energy efficiency and the QoS performance of data centers powered by our proposed BTVM strategy. We shed a bright light on SAR's strength in guaranteeing QoS requirements of the key workloads by the virtue of shortened workload latency. We illustrate how ESA effectively minimizes the total energy consumption caused by computing nodes and the cooling system under bursty workloads. To demonstrate the superiority of BTVM, we

Algorithm 3 : ESA for virtual machine placement

Input: $Iter, T_0, T_{end}, q$

Output: The optimal solution vector for the VMs to be placed on the servers;

```

1: Initial solution vector  $S$ ; //Randomly initialize the VMP solution;
2: Modulation initial solution  $S$ ; //Distribute a larger value to the servers on the top of the rack;
3:  $T = T_0$ ;
4: while  $T > T_{end}$  do
5:   for  $t \leftarrow 1$  to  $Iter$  do
6:      $Newsolution()$ ; // generate new placement solution vector;
7:      $Testsolution()$ ; // Test whether the new placement solution exchanges workloads of servers at the top of the rack, return 1 if this condition is true, otherwise, return 0;
8:     while  $Testsolution() == 1$  do
9:       Generate a new solution vector  $S$ ;
10:    end while
11:    Use the Metropolis Criterion to decide whether to accept the new solution vector  $S$ ;
12:    Update the solution vector  $S$ ;
13:  end for
14:   $T \leftarrow q \cdot T$ ; //Cooling process;
15: end while
16: return solution vector  $S$ 

```

compare BTVM with the three state-of-the-art algorithms PPVMP, XINT-GA, PLSC, GRANITE.

We carry out all the simulated experiments on a server equipped with an Intel(R)Core i7- 6700@3.4GHz and 8GB RAM. We build our *BTVM* strategy on MATLAB 2016a. After a microscopic analysis of a data set from the Alibaba data center [47], we intercept a 60-hour segment as our experimental dataset. And the overload threshold is set to 1400 for workload condition determination (see also Fig. 5). A variety of VMP schemes have been proposed to deal with bursty workloads. For this reason, we compare BTVM against four schemes, namely, FCFS, PPVMP, XINT-GA, PLSC, and GRANITE. Now, we briefly introduce these schemes as follows before delving into performance comparisons.

GRANITE (a GReedy based scheduling Algorithm miNimizing Total Energy) [18]: It is a VM scheduling algorithm for lowering holistic data center energy consumption. GRANITE is divided into two phases, i.e., the virtual machine placement phase (VMP) and the virtual machine scheduling phase. In the VMP phase, a greedy algorithm is used to find the optimal placement scheme. In the migration phase, a greedy algorithm is employed to select VMs on servers with temperatures exceeding a specified threshold for migration.

FCFS (First Come First Serve) [8]: FCFS serves as a baseline strategy for the experimental evaluation. Workloads without being partitioned are processed in a first-come-first-serve manner.

PPVMP (Power-Aware and Performance-Guaranteed Virtual Machine Placement) [21]: PPVMP is ant-colony-

optimization-based VMP, which strives to minimize physical machine power consumption and guarantees VM performance. As a bi-objective VMP, PPVMP builds a power consumption model to estimate the energy cost of active servers. PPVMP also incorporates a virtual machine performance model to evaluate the performance degradation of VMs incurred by resource competition and sharing.

XINT-GA (Genetic Algorithm) [36]: GA, a meta-heuristic iterative algorithm, uses crossover, mutation, and selection operations to seek solutions to optimization problems. The extension of GA – XINT-GA – was developed to distribute incoming workloads on different servers to maximize supply temperature and minimize cooling costs under peak inlet temperature constraints.

PLSC (Peak Load Scheduling Control Method) [3]: PLSC improves energy efficiency of data centers by promoting the QoS of peak loads. PLSC integrates two request queues PQ and SQ. PLSC preferentially schedules critical requests to PQ to ensure QoS. Delay-tolerant requests are then scheduled to the SQ with lower priorities. In addition, PLSC advocates for a hardware resource allocation strategy to optimize load scheduling on the energy consumption of data centers.

A. Experimental Setups

The simulator we build is derived from Flovent, which is an easy to use simulation software. A 42U industry-standard data center contains two rows of industry-standard racks in the simulator. Many of our previous research efforts have used this simulator [36]. We conduct extensive experiments via a real-world data center simulator to gauge the effectiveness of BTVM in terms of workload delay and energy efficiency, and we have the following simulation configurations throughout this empirical study [26]. The $9.6m \times 8.4m \times 3.6m$ simulated data center comprises two rows of industry-standard 42U racks. Each rack has five chassis (labeled as A, B, C, D, and E from bottom to top in Fig. 4). And each chassis contains 10 computing nodes; each computing node is equipped with 10 processors and 16 GB RAM. To cool down computing nodes, each rack row is placed between a typical cold and hot aisle. And the air speed of the cold air flow provided by the cooling system is $8m^3/s$. Since each VM is set to consist of 2 processors and 4 Gb RAM, the total number of VMs that the simulated data center needs to process is derived as $10 * 5 * 10 * 8/2 = 2000$ VMs. Moreover, the model of all the computing nodes of the simulated data center is DELL PowerEdge 1855. According to ASHRAE's standard range of air cold temperatures and the supply temperature values expounded in [48], we set the supply temperature from CRAC 17 °C. The peak threshold temperature of the data center is set as 26 °C [45]. The cross-interference coefficient matrix A and the heat distribution matrix D of this data center are reported in the literature [36]. The simulator has been open sourced at the following GitHub address: <https://github.com/dscLabJNU/Heat-Recirculation-Aware-VMP-Matlab->.

B. Performance Evaluation

We quantitatively compare BTVM with XINT-GA, PPVMP, and PLSC in terms of computing power consumption,

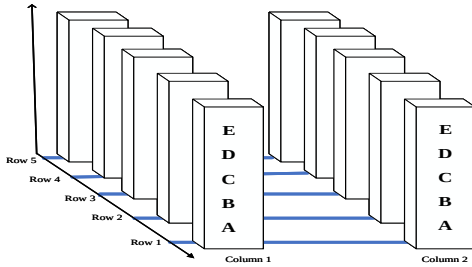


Fig. 4: Outline drawing of a small-scale data center referenced in [45]

cooling power consumption, total energy cost, and execution time. We further assess the performance of the enhanced SA algorithm or ESA.

1) Performance Evaluation between SA and ESA

We set the number of requested VMs in a range between 400 and 1600 [45] with an intent to investigate the performance of the ESA algorithm. Fig. 5 sketches the frequency of iterations of ESA compared to SA. The average number of iterations attained by the ESA algorithm is 39, whereas the number of iterations obtained by the SA algorithm is 57. It is apparent that the ESA algorithm optimizes the number of iterations by 32% as opposed to the SA alternative. Consequently, we choose ESA as the VMP algorithm to achieve the fastest execution time with the same optimized values.

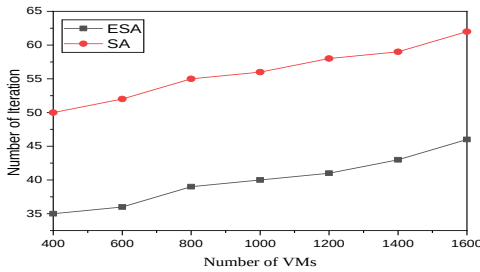
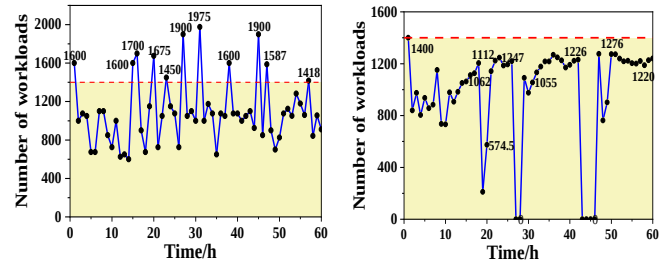


Fig. 5: Compared of Frequency of Iterations

2) Workload Delay

In this group of experiments, we evaluate the QoS trait of the BTVMP-powered data center under bursty workloads. We start the experiments by examining the execution status of workloads in the simulated data center equipped with BTVMP. Fig. 6(a) shows the number of workloads executed by BTVMP over a 60-minute period, and Fig. 6(b) reveals that the number of processed workloads never exceeds the predefined workload threshold of 1400. This finding is expected because BTVMP intentionally prioritizes key workloads or delay-sensitive workloads while delaying low-priority workloads that exceed the QoS threshold: BTVMP successfully guarantees the performance of data centers under bursty loads.

Apart from the above study on the workload execution status in BTVMP, we categorize the execution time of all incoming workloads based on a large number of trace analyses [3] [8]. Fig. 6 unravels statistics on the execution time of classified workloads: Fig. 7(a) plots the time-varying proportions of workloads, and Fig. 7(b) depicts the percentages of three

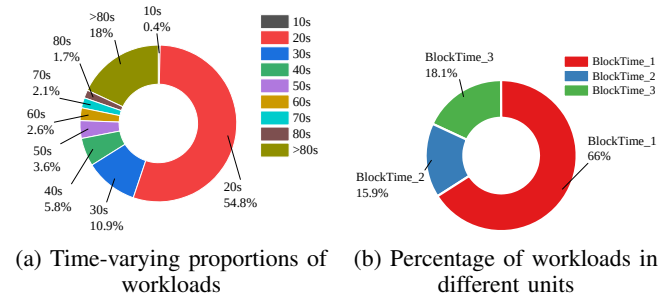


(a) Fluctuation in the number of workloads within 60 hours

(b) The situation after the workload is processed

Fig. 6: Before and after the workload execution status in BTVMP

workload categories. A workload with an execution time of 30 seconds is defined as **BlockTime_1**; a workload whose execution time is between 30 and 80 seconds is defined as **BlockTime_2**; a workload that takes more than 80 seconds to execute is defined as **BlockTime_3**. Fig. 7(b) reveals that workloads **BlockTime_1**, **BlockTime_2**, and **BlockTime_3** account for 66%, 15.9%, and 18.1% of the total workloads, respectively. We assume that a delay sensitivity threshold of 30s is defined. **BlockTime_1** satisfies the delay-sensitivity workload requirement.



(a) Time-varying proportions of workloads

(b) Percentage of workloads in different units

Fig. 7: Statistics on the execution time of classified workloads.

As the competitors PPVMP and XINT-GA that ignore the burstiness problem, our BTVMP strategy is compared with FCFS and PLSC in terms of average workload delay to make fair comparisons. Fig. 8 displays the average workload delay of BTVMP, PLSC, FCFS with respects to the three workload categories **BlockTime_1**, **BlockTime_2**, and **BlockTime_3**.

Fig. 8 (a) shows the average delay of BTVMP, PLSC and FCFS under ordinary workload conditions. We observe from Fig. 8(a) that BTVMP outperforms PLSC and FCFS in the case of **BlockTime_1** workloads with an execution time of 30 seconds. For instance, the average workload delay of BTVMP is 18% and 48% of those of PLSC and FCFS, respectively. BTVMP yields a slightly longer average workload delay in **BlockTime_3** workloads that take more than 80 seconds to execute. This result is acceptable because the proportion of **BlockTime_3** workloads is small, and clients with long-execution-time workloads are less sensitive to workload delay. It is evident that BTVMP has impressively lower latency of ordinary workloads to fend off the long-tail effect induced by

ordinary-workload response latency.

Fig. 8(b) shows the average delay of BTVM, PLSC, and FCFS when key workloads are running. Fig. 7(b) indicates that BTVM sits between PLSC and FCFS in terms of workload delay. In particular, the average workload delay of BTVM is slightly higher than that of PLSC but BTVM wins over FCFS by a large margin.

In case of workload bursts, service providers assign high priorities to key workloads, but it is more likely to cause the undesired long tail effect for ordinary workloads. Fortunately, our BTVM is capable of prioritizing critical workloads while preventing low-priority workloads from starvation. In summary, these experimental results demonstrate that BTVM satisfies the QoS requirements of key workloads while curtailing the delay of ordinary workloads at the acceptable cost of key workload delays.

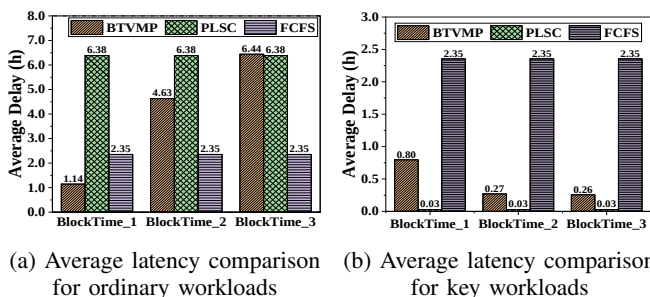


Fig. 8: Comparison of the average workload delay

3) Energy Consumption of Servers and Cooling.

The tested system, driven by a trace gleaned from real-world data center workloads in the Alibaba Cloud [47] over a 75-day period, manages and processes a total of 38 million requests. We adjust the workloads by intercepting 60 hours of volume, and the 60-hour load is converted into the number of VMs required per hour as input parameters for the experiments. Each algorithm, given the experimental parameter setup, governs the VM workloads once per hour to place the number of arriving VM tasks. VMs that are already running continue to execute until the resources are reclaimed. We assume that the placed servers remain unchanged; otherwise, VM scheduling will be involved. The placed VMs run in the server until the end of application lifetime. In the experiment, we set the period as one hour. Unlike other energy consumption components in computing resources, the energy consumption of cooling systems has a late response to control actions. The behavior of cooling control systems is not an immediate response to the actuator at the thermal level. In other words, the cooling systems take a certain period, which cannot be too short, to adjust the temperatures of data centers. And a large period will result in a long wait time of workloads with deteriorated execution efficiency. Therefore, this period is normally configured as one hour. We measure the performance of BTVM as well as the alternatives – XINT-GA, PPVMP, PLSC, and GRANITE – in terms of energy consumption of computing nodes and the cooling system (see Fig. 9).

Servers Energy Consumption Fig. 9 depicts the energy consumption of servers managed by BTVM, XINT-GA,

PPVMP, PLSC, and GRANITE in a 60-minute period. The results reveal that XINT-GA causes the highest energy consumption of servers due to XINT-GA's lack of awareness of energy efficiency in computing nodes: XINT-GA keeps all servers in the active mode even during light workloads. The findings unveil that BTVM effectively conserves the energy consumption of servers in data centers. Specifically, PPVMP and our BTVM have a clear edge over PLSC and XINT-GA in terms of energy efficiency. For example, BTVM cuts back the average node energy consumption of XINT-GA and PLSC by 11.5% and 2.8%, respectively. And the average server energy consumption of BTVM is lower than that of PPVMP by a small margin of 2.8%, which confirms that BTVM is adroit at optimizing server energy efficiency. Additionally, BTVM saves 1.018% of the average server energy consumption compared to the GRANITE. These results are expected because PPVMP is an ACO-based algorithm dedicated to minimizing the energy consumption of servers, whereas our BTVM's focal point is in optimizing the cooling energy consumption of data centers.

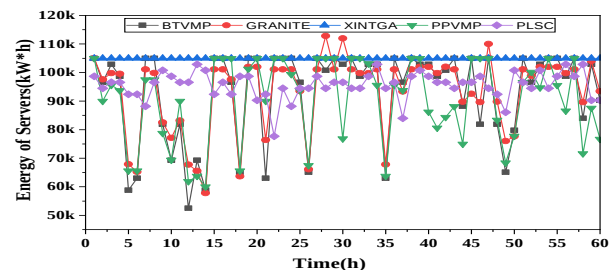


Fig. 9: Comparison of the computing energy consumption among different algorithms

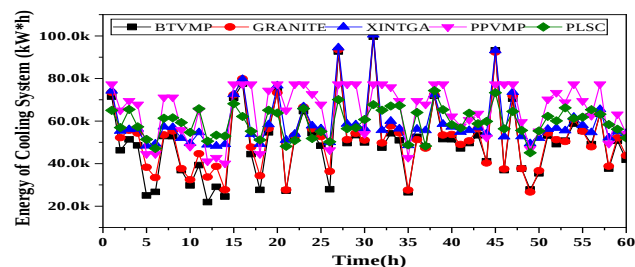


Fig. 10: Comparison of cooling energy consumption among different algorithms

Cooling Energy Consumption. Now we examine, Fig. 10 shows that BTVM outperforms XINT-GA, PPVMP, PLSC, and GRANITE with respect to cooling performance. The reason behind BTVM's cooling efficiency advantage is that BTVM judiciously mitigates the heat-recirculation effect when VMP decisions are made. Such maneuvers in heat-recirculation effect enables BTVM to place more VMs on computing nodes close to a cooling system, thereby avoiding hot spots and lowering cooling energy cost.

In contrast, PPVMP and PLSC overlook cooling energy issue, in which local hot spots may occur during VMP. Consequently, cooling systems are forced to operate at a lower

temperature, thus imposing a heavy burden on the cooling system and increasing the cooling energy consumption. XINT-GA, on the other hand, considers the heat-recirculation effect in data centers. However, XINT-GA places VMs evenly on all the servers. Though this VMP strategy reduces the probability of local hotspots, XINT-GA is inadequate for revamping the energy efficiency of cooling systems. The GRANITE algorithm adopts a greedy algorithm in the initialization and deployment phase of the VMs to obtain a placement solution for the VM workload, which does not result in an optimal placement solution, and this method also does not consider the maximum server utilization, which generates localized hotspots.

Thanks to the heat re-circulation reduction, BTVMP is in a leading position to lower the probability of arising hotspots - conserving cooling energy while enhancing the reliability of data centers. According to a prior work [45], we also confirm that the performance improvement of ESA deployed in BTVMP becomes more pronounced in large-scale applications during the virtual machine placement.

C. Total Energy Consumption and Execution Time

Now, we evaluate the total energy consumption and execution time of our BTVMP and its competitors – XINT-GA, PPVMP, PLSC, and GRANITE.

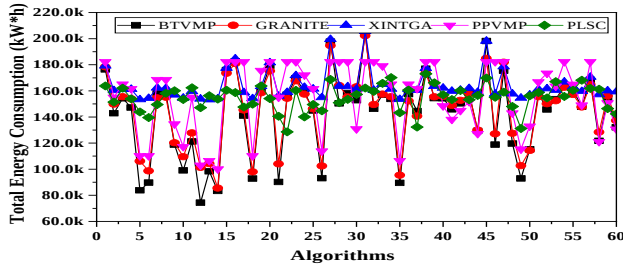


Fig. 11: Comparison of the total energy consumption among different algorithms

Total Energy Consumption Evaluation: Fig. 11 depicts the total energy consumption of BTVMP, XINT-GA, PPVMP, PLSC, and GRANITE. Not surprisingly, the results show that BTVMP achieves the highest energy efficiency among its VMP peers. For example, BTVMP bolsters the energy efficiency of XINT-GA, PLSC, and PPVMP by 13%, 7.9%, 8.4%, and 1.9 %, respectively. BTVMP's championship is attributed to its heat re-circulation reduction amid making VMP decisions, and this strategy, in turn, diminishes the adverse impact of hotspots on energy cost. PLSC can slash total energy consumption to some degrees, but PLSC ignores the crucial factor of cooling energy.

Execution Time Evaluation: In this part of the study, we probe the execution time for the five VMP schemes to assess their optimal VMP solutions. ESA algorithm is optimized based on simulated annealing algorithm (SA), which is an iterative algorithm. This algorithm does not have a fixed time complexity because for each execution, the number of iterations generated by obtaining the optimal solution is

different. But in this part, we compare the time required to execute different algorithms.

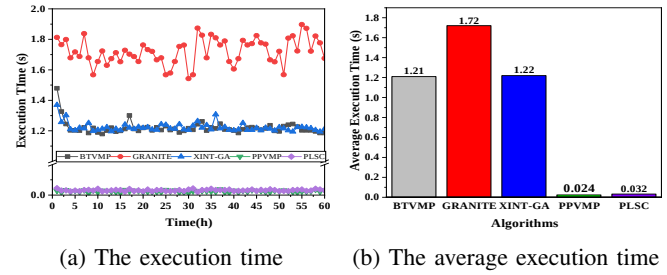


Fig. 12: Execution time of each algorithm

Fig. 12 plots the execution time of the five schemes BTVMP, XINT-GA, PPVMP, PLSC, and GRANITE. Fig. 12(a) shows that the execution time of our BTVMP is within a reasonable range. For example, the execution time of our BTVMP is higher than those of PPVMP and PLSC but lower than that of XINT-GA, GRANITE. When it comes to the average execution time, Fig. 12(b) shares a similar trend with Fig. 12(a), the findings of which are expected thanks to three factors. First, PPVMP merely focuses on placing workloads on a small number of active computing nodes, thereby costing less execution time. Second, BTVMP and XINT-GA are iterative-based optimization algorithms that give rise to a longer execution time than the PPVMP and PLSC cases. Third, PPVMP and XIN-GA either only optimize energy consumption of servers or give heed to cooling cost. The PLSC algorithm only deals with incoming peak workloads. GRANITE spends more execution time than the BTVMP algorithm because it considers the initial deployment and dynamic migration of VMs.

D. Confidence intervals and sensitivity analysis

In the context of an experimental evaluation presented in the paper, the BTVMP approach is suggested to optimize various performance metrics, including computing energy consumption, cooling energy consumption, total data center energy consumption, and the degree of Service Level Agreement (SLA) violations. Additionally, a new performance criterion called ESV is introduced, which takes into account multiple factors, such as energy consumption and SLA violations, for a comprehensive evaluation of system performance [42].

$$ESV = E_{Total} \times SLAV \quad (12)$$

A comparison of five different algorithms is made based on the ESV criterion in Fig. 13. BTVMP outperforms the PLSC, XINT-GA, PPVMP, and GRANITE algorithms in terms of energy savings. GRANITE takes into account the migration process of VMs, which results in reduced service performance of VMs during migration. PPVMP does not take into account the impact of heat recirculation in servers and hence does not achieve good results in optimizing the overall datacenter energy consumption. PLSC and XINT-GA algorithms achieve higher values for SLA violations. Overall, in terms of data

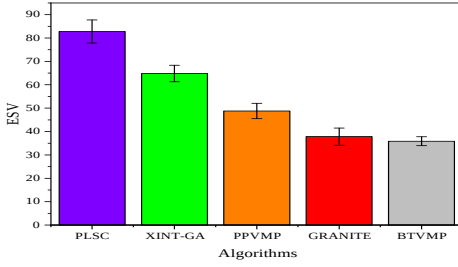


Fig. 13: Comparison of different algorithms based on ESV criterion for the Alibaba workload in the different 95% confidence interval

center energy optimization, EAVMS is highlighted as superior to all state-of-the-art methods. It performs well in ESV criterion evaluation and maintains a high level of confidence at 95%, which suggests that EAVMS is a robust solution for energy optimization in data centers.

In addition, we perform hyperparameter sensitivity analysis of significant parameters. We evaluate the impact on overall energy consumption and VM performance degradation under different data center utilization thresholds. In the context of Fig. 14, the server resource utilization threshold plays a critical role in simplifying the experimental procedure and optimizing the total energy consumption of a data center. This threshold is set between 40% and 90%, and its level has a significant impact on the overall energy usage. When the server resource utilization threshold is set too low, it necessitates the activation of more servers to handle the workload, resulting in excessively high energy consumption. Conversely, if the threshold is set too high, such as at 90%, it can lead to Service Level Agreement (SLA) violations. This occurs because the server's resources, including storage capacity, memory, and bandwidth, may not be sufficient to meet the demands of all virtual machines (VMs). To strike a balance between maintaining Quality of Service (QoS) and minimizing the data center's total energy consumption, the optimal server utilization threshold is found to be 80%. Consequently, this threshold level is selected for the experiments to achieve this equilibrium.

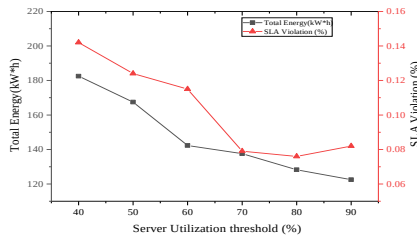


Fig. 14: Comparison of total energy and SLA violation under different server utilization thresholds

According to ASHRAE's standard range of air cold temperatures and the supply temperature values expounded in [48], The supply temperature from CRAC is set as 17-27 °C. In terms of cooling energy savings, BTVMP is capable of achieving cooling energy savings of 4.7% to 23.8% in comparison to GRANITE, XINT-GA, PPVMP and PLSC in the experimental

evaluation section. The supply temperatures optimized by our BTVMP strategy are shown in Fig. 15(a). According to Eq. 6, BTVMP is the most effective in optimizing the supply temperature, which optimized cooling range conforms to ASHRAE's standard. In addition, we set the peak threshold temperature for the server to 26°C according to reference [45], and the server inlet temperature should be within the peak temperature. Therefore, we experimentally demonstrate that the server inlet temperature variations caused by different periods of VM placement still fulfill within the peak threshold temperature. In Fig.15(b), it can be observed that our setting of a maximum peak temperature of 26 degrees Celsius is appropriate as the inlet temperature variations over time are all within the peak temperature, with only a few inlet temperatures violating the maximum peak temperature.

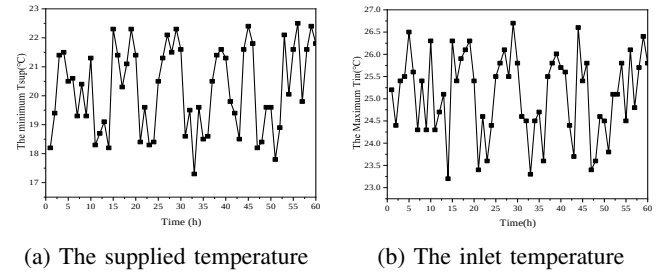


Fig. 15: The supply temperature T_{sup} and inlet temperature T_{in} optimized by BTVMP for workloads arriving in different periods

VI. CONCLUSION

In this paper, we proposed BTVMP - a burst-aware and thermal-efficient virtual machine placement approach for cloud data centers. BTVMP takes QoS and energy-efficiency into account amid making VMP decisions. To tackle bursty workloads, we designed a split-and-recombine algorithm coined as SAR - a scheme that prioritizes critical workloads and delays lower-priority workloads to ensure QoS requirements. To minimize the energy consumption of cloud data centers, we implemented an enhanced simulated-annealing-based heuristic algorithm - a thermal-aware ESA that optimizes virtual machine placement by mitigating the heat recirculation effect.

We conducted extensive experiments to evaluate the performance of BTVMP in terms of workload delay, energy efficiency, cooling cost, and execution time. We compared BTVMP with FSFC and the three other state-of-the-art algorithms XING-GA, PPVMP, PLSC, and GRANITE. The experimental results indicate that our BTVMP lowers the total energy consumption of the existing techniques by 27.8%-49.4%. Moreover, our findings demonstrate that BTVMP significantly promotes the QoS of data centers, especially under bursty workload conditions. BTVMP bolsters the average delay time efficiency of algorithms PLSC and FCFS by 18% and 11%, respectively.

We plan to expand our work in the following four aspects future research directions. First, we will further delve into a data center network topology to grapple with the bursty-workload issue. Second, we intend to improve the applicability

of BTVM in heterogeneous data centers by incorporating multifaceted factors such as computing node and VMs types. Third, after the implementation of the VM placement technique, we are positioned to explore a real-time VM migration strategy as a companion module of BTVM. Fourth, the proposed approach will be extended to manage energy-efficient data storage systems.

ACKNOWLEDGMENT

This work is sponsored by the National Natural Science Foundation of China under Grant No.62072214, Guangdong Basic and Applied Basic Research Foundation under Grant No.2021B1515120048, the Science and Technology Planning Project of Guangzhou under Grant 202103000036, the Open Project Program of Wuhan National Laboratory for Optoelectronics No.2020WNLOK006, and the Industry-University-Research Collaboration Project of Zhuhai under Grant No.ZH22017001210048PWC. Xiao Qin's work is supported by the U.S. National Science Foundation under Grants IIS-1618669 and OAC-1642133. The corresponding author of this paper is Yuhui Deng.

REFERENCES

- [1] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, "A priority, power and traffic-aware virtual machine placement of iot applications in cloud data centers," *Journal of Systems Architecture*, vol. 115, p. 101996, 2021.
- [2] K. Gai, X. Qin, and L. Zhu, "An energy-aware high performance task allocation strategy in heterogeneous fog computing environments," *IEEE Transactions on Computers*, vol. 70, no. 4, pp. 626–639, 2021.
- [3] C. Hu, Y. Deng, G. Min, P. Huang, and X. Qin, "Qos promotion in energy-efficient datacenters through peak load scheduling," *IEEE Transactions on Cloud Computing*, vol. 9, no. 02, 2021.
- [4] S. Deng, C. Zhang, C. Li, J. Yin, S. Dustdar, and A. Y. Zomaya, "Burst load evacuation based on dispatching and scheduling in distributed edge networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 8, pp. 1918–1932, 2021.
- [5] J. Wang, X. Wang, T. Shen, Y. Wang, L. Li, Y. Tian, H. Yu, L. Chen, J. Xin, X. Wu *et al.*, "Parallel vision for long-tail regularization: Initial results from ivfc autonomous driving testing," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 286–299, 2022.
- [6] R. F. da Silva, T. Glatard, and F. Desprez, "Self-healing of workflow activity incidents on distributed computing infrastructures," *Future Generation Computer Systems*, vol. 29, no. 8, pp. 2284–2294, 2013.
- [7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
- [8] L. Lu, P. Varman, and K. Doshi, "Graduated qos by decomposing bursts: Don't let the tail wag your server," in *Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems*, 2009, p. 12–21.
- [9] K. Jang, J. Sherry, H. Ballani, and T. Moncaster, "Silo: Predictable message latency in the cloud," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, p. 435–448.
- [10] T. Zhu, M. A. Kozuch, and M. Harchol-Balter, "Workloadcompactor: Reducing datacenter cost while providing tail latency slo guarantees," in *Proceedings of the 2017 Symposium on Cloud Computing*, 2017, p. 598–610.
- [11] M. C. Calzarossa, L. Massari, and D. Tesserà, "Workload characterization: A survey revisited," *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, pp. 1–43, 2016.
- [12] R. T. Kaushik, L. Cherkasova, R. Campbell, and K. Nahrstedt, "Lightning: Self-adaptive, energy-conserving, multi-zoned, commodity green cloud storage system," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, 2010, p. 332–335.
- [13] Y. Liu, R. Gunasekaran, X. Ma, and S. S. Vazhkudai, "Automatic identification of application i/o signatures from noisy server-side traces," in *Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST 14)*, 2014, pp. 213–228.
- [14] S. Ilager, K. Ramamohanarao, and R. Buyya, "Thermal prediction for efficient energy management of clouds using machine learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1044–1056, 2021.
- [15] Y. Ran, X. Zhou, H. Hu, and Y. Wen, "Optimizing data center energy efficiency via event-driven deep reinforcement learning," *IEEE Transactions on Services Computing*, vol. 16, no. 2, pp. 1296–1309, 2023.
- [16] C. Hu, Y. Deng, L. T. Yang, and Y. Zhao, "Estimating the resource demand in power-aware clusters by regressing a linearly dependent relation," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 3, pp. 385–397, 2021.
- [17] B. Guenter, N. Jain, and C. Williams, "Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 1332–1340.
- [18] X. Li, P. Garraghan, X. Jiang, Z. Wu, and J. Xu, "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 06, pp. 1317–1331, 2018.
- [19] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 732–794, 2015.
- [20] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomput*, vol. 60, no. 2, p. 268–280, 2012.
- [21] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, and Q. Zheng, "Power-aware and performance-guaranteed virtual machine placement in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, no. 6, pp. 1385–1400, 2018.
- [22] S. Rahmani, V. Khajehvand, and M. Torabian, "Burstiness-aware virtual machine placement in cloud computing systems," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 362–387, 2020.
- [23] Z. Luo and Z. Qian, "Burstiness-aware server consolidation via queuing theory approach in a computing cloud," in *Proceedings of the 27th IEEE International Symposium on Parallel and Distributed Processing*, 2013, pp. 332–341.
- [24] Z. Li, X. Yu, and L. Zhao, "A strategy game system for qos-efficient dynamic virtual machine consolidation in data centers," *IEEE Access*, vol. 7, pp. 104 315–104 329, 2019.
- [25] Z. A. Mann, "Multicore-aware virtual machine placement in cloud data centers," *IEEE Transactions on Computers*, vol. 65, no. 11, pp. 3357–3369, 2016.
- [26] H. Feng, Y. Deng, and J. Li, "A global-energy-aware virtual machine placement strategy for cloud data centers," *Journal of Systems Architecture*, vol. 116, p. 102048, 2021.
- [27] X.-F. Liu, Z.-H. Zhan, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 113–128, 2018.
- [28] M. Xu, A. N. Toosi, and R. Buyya, "A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 4, pp. 544–558, 2020.
- [29] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *Proceedings of the 1st International Conference on Cloud Computing*, 2009, p. 254–265.
- [30] A. Beloglazov and R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1366–1379, 2013.
- [31] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "Grvm: A greedy randomized algorithm for virtual machine placement in cloud data centers," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2571–2582, 2021.
- [32] S. Azizi, M. Zandsalimi, and D. Li, "An energy-efficient algorithm for virtual machine placement optimization in cloud data centers," *Cluster Computing*, no. 4, p. 3421–3434, 2020.
- [33] C. Guo, K. Xu, G. Shen, and M. Zukerman, "Temperature-aware virtual data center embedding to avoid hot spots in data centers," *IEEE Transactions on green communications and networking*, vol. 5, no. 1, pp. 497–511, 2020.
- [34] S. Ilager, K. Ramamohanarao, and R. Buyya, "Etas: Energy and thermal-aware dynamic virtual machine consolidation in cloud data center with

proactive hotspot mitigation,” *Concurrency and Computation: Practice and Experience*, vol. 31, no. 17, p. e5221, 2019.

- [35] R. Chen, B. Liu, W. Lin, J. Lin, H. Cheng, and K. Li, “Power and thermal-aware virtual machine scheduling optimization in cloud data center,” *Future Generation Computer Systems*, vol. 145, pp. 578–589, 2023.
- [36] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, “Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 11, pp. 1458–1472, 2008.
- [37] J. Li, Y. Deng, Y. Zhou, Z. Zhang, G. Min, and X. Qin, “Towards thermal-aware workload distribution in cloud data centers based on failure models,” *IEEE Transactions on Computers*, vol. 72, no. 2, pp. 586–599, 2023.
- [38] H. Yuan, J. Bi, and M. Zhou, “Spatiotemporal task scheduling for heterogeneous delay-tolerant applications in distributed green data centers,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1686–1697, 2019.
- [39] N. Wang, J. Zhang, and X. Xia, “Energy consumption of air conditioners at different temperature set points,” *Energy and Buildings*, vol. 65, pp. 412–418, 2013.
- [40] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, “Making scheduling ‘cool’: Temperature-aware workload placement in data centers,” in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ser. ATEC ’05, 2005, p. 5.
- [41] J. Zeng, D. Ding, K. Kang, H. Xie, and Q. Yin, “Adaptive drl-based virtual machine consolidation in energy-efficient cloud data center,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2991–3002, 2022.
- [42] J. Ahmadi, A. Toroghi Haghighat, A. M. Rahmani, and R. Ravanmehr, “Confidence interval-based overload avoidance algorithm for virtual machine placement,” *Software: Practice and Experience*, vol. 52, no. 10, pp. 2288–2311, 2022.
- [43] N. Metropolis, “Equation of state calculations by fast computing machines,” *Journal of Chemical Physics*, vol. 21, 1953.
- [44] R. Rutenbar, “Simulated annealing algorithms: an overview,” *IEEE Circuits and Devices Magazine*, vol. 5, no. 1, pp. 19–26, 1989.
- [45] H. Feng, Y. Deng, Y. Zhou, and G. Min, “Towards heat-recirculation-aware virtual machine placement in data centers,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 256–270, 2021.
- [46] İlhan İLHAN, “An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem,” *Swarm and Evolutionary Computation*, vol. 64, p. 100911, 2021.
- [47] <https://github.com/alibaba/clusterdata>.
- [48] ASHRAE, “American society of heating, refrigerating and air conditioning engineers,” <http://tc0909.ashraetcs.org/>, 2018.

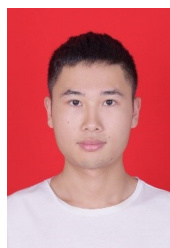


Jie Li received the B.E. degree in automation from Wanjia University of Technology, Anhui, China, in 2016, and the MS degree in computer science and technology from the Guangxi University for Nationalities, Nanning, China, in 2019. He is a Teacher with the School of Software, South China Normal University. He has published several papers in IEEE TC, IEEE TCC, IEEE TNSM, HPCC, ACM PACT, etc. His current research interests include parallel and distributed computing, data center architecture, cloud computing and data replica placement.



Yuhui Deng received the Ph.D. degree in computer science from the Huazhong University of Science and Technology, in 2004. He is a professor with the Computer Science Department, Jinan University. Before joining Jinan University, he worked at EMC Corporation as a senior research scientist from 2008 to 2009. He worked as a research officer at Cranfield University in the United Kingdom from 2005 to 2008. His research interests cover data management, data analysis, data storage, cloud computing, computer architecture, performance evaluation, etc. He has published over 100 scientific papers including ACM Computing Surveys, ACM TOS, IEEE TPDS, IEEE TC, IEEE TCAD, IEEE TBD, IEEE TCC, IEEE TNSM, ACM TDS, ICCD, DATE, MASCOTS, etc. He was on the finalist of storage challenge in Supercomputing 2007, and the finalist of EMC global innovation showcase in 2008. He was awarded annual best paper award from JISE in 2017, and best journal paper award of IEEE TBD from the Big Data Technical Committee of IEEE Communications Society in 2019.

Authorized licensed use limited to: Auburn University. Downloaded on April 09, 2024 at 22:00:38 UTC from IEEE Xplore. Restrictions apply.



Rui Wang received the MS degree in information science and technology from Jinan University, Guangzhou, China, in 2021. His current research interests include data center architecture, cloud computing and data provenance.



Yi Zhou received the B.E.E., M.S.E.E. degrees in electronic engineering all from Beijing University of Technology, Beijing, in 2006 and 2010 respectively. He received the PhD degree in computer science from Auburn University in 2018. He is currently an assistant professor in TSYS School of Computer Science at Columbus State University. Prior to joining Columbus State University in 2018, he has been a software engineer in Alcatel-Lucent Technologies (China) Co., Ltd. for four years from 2010 to 2014. His research interests include energy-saving techniques, database systems, big data techniques and parallel computing.



Hao Feng is a teacher in School of Computer and Cyberspace Security, Hainan University. He received the Ph.D. degree in computer application technology from the Computer Science Department of Jinan University. He received the B.E. degree in computer science and technology from Dalian Maritime University in 2014, and the MS degree in computer science and technology from the Guangxi University in 2017. His current research interests include parallel and distributed computing, data center architecture, cloud computing and resources management.



Geyong Min is a Professor of High Performance Computing and Networking in the Department of Mathematics and Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Future Internet, Computer Networks, Wireless Communications, Multimedia Systems, Information Security, High Performance Computing, Ubiquitous Computing, Modelling and Performance Engineering. He serves as an Associate Editor of IEEE Transactions on Cloud Computing, IEEE Transactions on Sustainable Computing, IEEE Transactions on Computers, and etc.



Xiao Qin received the BS and MS degrees in computer science from the Huazhong University of Science and Technology, Wuhan, China, in 1992 and 1999, respectively, and the PhD degree in computer science from the University of Nebraska-Lincoln, Lincoln, Nebraska, in 2004. He is currently an alumni professor and the director of Graduate Programs with the Department of Computer Science and Software Engineering, Auburn University. He won an NSF CAREER Award, in 2009. His research interests include parallel and distributed systems, storage systems, fault tolerance, real-time systems, and performance evaluation.

Authorized licensed use limited to: Auburn University. Downloaded on April 09, 2024 at 22:00:38 UTC from IEEE Xplore. Restrictions apply.