Towards Thermal-Aware Workload Distribution in Cloud Data Centers Based on Failure Models

Jie Li, Yuhui Deng, Yi Zhou, Zhen Zhang, Geyong Min, and Xiao Qin

Abstract—Increasing workload conditions lead to a significant surge in power consumption and computing node failures in data centers. The existing workload distribution strategies focused on either thermal awareness or failure mitigation, overlooking the impact of node failures on the energy efficiency of cloud data centers. To address this issue, a new holistic model is built to characterize the impacts of workloads, computing and cooling costs, heat recirculation, and node failure on the energy efficiency of cloud data centers. Leveraging such a holistic model, we propose a novel thermal-aware workload distribution strategy called *HGSA* that takes node failure into accountand can improve the energy efficiency of cloud data centers. Our empirical findings confirm that (i) faulty nodes lead to a large rise in power consumption, and (ii) failure locations play a vital role in the power consumption of data centers. Experimental results unveil that HGSA is adroit at making near-optimal decisions in workload distribution strategies. In particular, HGSA cuts down the minimum inlet temperature by 5.2%-15%, improves the maximum air temperature of a Computer Room Air Conditioner (CRAC) model by 4.2%-26.5%, lowers the cooling cost by 15.4%-50% compared to the existing solutions. Furthermore, HGSA cuts back the total power consumption by 0.65%-78%.

Index Terms—Data Centers, Power Consumption, Node Failure, Energy Efficiency, Workload Distribution, Thermal-Aware.

1 Introduction

THERMAL-aware workload distribution schemes offer energy-efficient solutions to data centers. Traditional thermal-aware workload distribution policies are anchored on a model of well-functioning nodes, failing to consider two indispensable aspects - node failure and cooling costs. To fill this gap, we propose a novel thermal-aware workload distribution strategy by taking into account node failure as well as the total power consumption costs. Our overarching goal is to minimize peak inlet temperatures across all computing nodes, thereby lowering the total power consumption of data centers. At the heart of our workload distribution strategy is a holistic thermal model that integrates the impacts of node failure. We propose the *HGSA* algorithm - a heuristic and hybrid algorithm based on *Genetic Algorithm* (GA) and *Simulated Annealing* (SA) - to optimize workload distribution on computing nodes. Governed by our holistic model, HGSA redistributes workloads from faulty nodes to active nodes for processing. The following three motivations make our failure-based thermal-aware workload distribution strategy desirable and achievable.

1) Node failures become an indispensable factor for the development of cost-effective data centers.
2) Existing models built upon non-failure assumptions are inadequate for thermal-aware workload distribution.
3) There is a pressing demand to conserve the energy cost of data centers.

Reliable computing is rooted in the perspectives and understandings of node failures. For example, characterizing failures help to evaluate the availability of clusters in data centers with respect to resource allocation [1]. Over the last decades, computing and storage services provided by cloud data centers spew out. Services in critical areas such as finance, transportation, telecom, and military must be available for 7x24 hours with a requirement of minimal maintenance. Node failures in data centers offering critical services will cause a tragic loss. Many factors contribute to node failures in modern data centers. Hardware failures are a key form of failure in data centers. Specifically, heat dissipation of the CPUs and disk drives of computing nodes becomes a primary concern. The imbalance of heat dissipation will lead to breakdowns of computing nodes. Besides, sophisticated and ubiquitous software applications running on computing nodes are prone to malfunctions, where resource utilization plays a vital role in the failure rate and power consumption of data centers [2].

An increasing number of modeling techniques were devised to facilitate the management and planning of computing nodes in data centers. With the explosive growth of computing nodes in data centers, most of the existing models are inadequate for modern data centers because the models are focused on either reliability and availability or improvements based on non-failure conditions [3] [4] [5] [6]. Unlike the prior studies, we pay attention to constructing a node-failure-based model that sheds light on the relationship between workloads and computing node fail-

- *J. Li, Y. Deng (corresponding author) and Z. Zhang are with the Department of Computer Science, Jinan University, Guangzhou, China, 510632. Y. Deng is also with Wuhan National Laboratory for Optoelectronics, Wuhan, China. E-mail: lijiegxmd11@163.com, tyhdeng@jnu.edu.cn, zzhang@jnu.edu.cn.*
- *Y. Zhou is with TSYS School of Computer Science, Columbus State University, Columbus, GA, 31097, 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.*

ures. Powered by such a model, our workload distribution strategy is able to thermal-efficiently redistribute workloads from faulty nodes to active nodes.

Ever-increasing energy costs become one of the predominant components of operational costs in data centers. Modern data centers are comprised of thousands of interconnected nodes devour a large amount of energy for providing various services, which leads to a continuous increase in global carbon dioxide emissions. Recent studies forecast that from 2011 to 2035, the power consumption of global data centers will increase by 66% [7]. Besides, the power consumption of cooling systems accounts for approximately 50%, becoming a huge fraction of the total power consumption of data centers [8]. Furthermore, node failures lead to a remarkable decrease in the supplied temperature to node inlets, which in turn exacerbates cooling power consumption. Therefore, it is imperative to take node failure into account when designing a thermal-aware workload distribution strategy for reducing cooling costs in data centers.

Motivated by the aforementioned three observations, we propose a thermal-aware workload distribution to reduce the power consumption of data centers by virtue of a holistic failure model. Our overarching goal is to redistribute the workloads from faulty nodes to active computing nodes in a thermal-efficient manner. We explore the impact of node failures on the power consumption of data centers. The major contributions of this work are summarized as follows.

- We propose a holistic thermal model, which takes into account temporal–spacial distribution of node failures. This novel model provides guidelines to make thermal-friendly workload distribution decisions.
- We develop *HGSA* - a hybrid algorithm based on the *GA* and *SA* to optimize workload distribution, aiming to cut down the total power consumption of data centers.
- We shed light on the impact of failure location on the inlet temperature of computing nodes. Experimental results confirm that compared with the existing solutions, our HGSA algorithm can significantly improve the energy efficiency of data centers.

The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 models data centers in a holistic way with detailed discussions and derivations. The design and implementation of the workload distribution algorithm HGSA are presented in Section 4. Section **??** presents the performance evaluation. Finally, in Section 6, we draw the conclusions along with future research directions.

2 Related Work

Essential factors such as thermal efficiency, operational energy efficiency, and cooling costs must be considered when making decisions in workload distribution as well as resource management to reduce the power consumption in data centers. Moreover, because node failure leads to numerous energy waste, it is also indispensable to take into account the impacts of node failure on both the computing environment and the cooling system of a data center. Though various failure models and thermal management techniques have been reported in the literature, most of these schemes still deliver unsatisfied energy efficiency. In this section, we pinpoint their design principles as well as their performance issues.

2.1 Node Failure Analysis and Prediction

Researchers embarked on inspiring studies to improve the energy efficiency of data centers by modeling node failures. For example, Schroeder *et al.* [15] analyzed node failures by investigating a data collection from two large high-performance computing sites. They advocated that failure time and recovery time are well modeled by the Weibull distribution and the lognormal distribution, respectively. Garraghan *et al.* built three online models to examine combinations of feature sets and techniques with the purpose of lowering computational overhead [16]. Sharma *et al.* presented a failure-aware VM consolidation method to improve the energy efficiency of data centers [17]. After exploring the cluster models with a bisection-based partitioning algorithm, Wang *et al.* investigated failure characteristics from the dimensions of time, space, product line, and components by looking into a four-year hardware failure of data centers [18]. Alquran *et al.* presented an analysis of network-partitioning failures in cloud computing systems [19]. A test framework called *NEAT* was built to characterize failure with ordering, timing, and network fault-tolerant. Chalermarrewong *et al.* developed a failure prediction method called *ARMA* for data centers. ARMA relies on a fault tree analysis to improve its prediction accuracy [5]. However, these temperature prediction-based methods lack scalability and suffer from low accuracy, long calculation time, and high complexity. This is because they only consider the failure prediction technology, and do not consider the load balancing problem after the failure occurs.

2.2 Thermal-Aware Workload Distribution in Data Centers

Increased temperature of computing nodes causes a large amount of power consumption in data centers. Meantime, hotspots give rise to a significant increase in the inlet temperature of host nodes, which exacerbates the energy problem and further pushes up the cooling costs of data centers. To deal with this problem, a wide variety of thermal-aware techniques were developed to reduce the power consumption of data centers. For example, Moreno *et al.* built a workload model to improve resource management and operational conditions of a cloud computing environment [20]. Shashikant *et al.* [3] proposed a gradient boosting-based machine learning algorithm to model the peak temperature of computing nodes, aiming to minimize the power consumption of computing nodes for data centers by suppressing nodes' peak temperature. The performance evaluation involves inlet temperature and total energy consumption. Li *et al.* [13] built an elaborate thermal model, and proposed a Virtual Machine (VM) scheduling algorithm called GRANITE to lower the total energy consumption of data centers. GRANITE algorithm is responsible for Virtual Machine Placement (VMP) and migration based on the models identified off-line, consisting of a workload model, a

TABLE 1: Comparisons between HGSA and the other existing schemes

Schemes	Computing nodes modeling	Thermal awareness	Node failure modeling	Workload distribution modeling	Cooling cost consideration
MPIT-TA [9]	✗	✓	✗	✓	✓
TP-Model [3]	✓	✓	✗	✓	✓
TAW-Placement [4]	✗	✓	✗	✓	✓
ARMA [5]	✓	✗	✗	✓	✗
ITA [10]	✗	✓	✗	✓	✗
OLP-Model [11]	✓	✗	✗	✓	✗
AF-Model [2]	✗	✓	✓	✓	✗
TA-Distribution [12]	✗	✓	✗	✓	✗
GRANITE [13]	✓	✓	✗	✓	✓
PPVMP [14]	✓	✗	✗	✓	✗
HGSA	✓	✓	✓	✓	✓

computing node model, and a cooling model. Unfortunately, their performance evaluation does not take into account the failures of computing nodes. Zhao *et al.* [14] investigated the close tie between energy consumption and CPU utilization to design a non-linear power model, and developed a power-aware and performance-guaranteed VMP (PPVMP) algorithm to keep the balance between saving computing node energy and guaranteeing VM performance. The theoretical model is composed of a computing node power consumption model and a VM performance model. However, it does not mention the impact of heat recirculation among active computing nodes, and it also does not consider the impact of VMP on cooling cost. Furthermore, their performance only considers the power consumption of computing nodes and the performance of VM. Tang *et al.* [9] developed a thermal-aware workload scheduling strategy, where workloads are placed on computing nodes with the highest cooling efficiency, thus, lowering the impact of heat-recirculation among computing nodes and improving the cooling efficiency. The theoretical model involved is composed of computing power, cooling cost, total power consumption, and inlet temperature. However, the impacts of node failure and failure locations on energy consumption are not considered. The performance evaluation includes computing power, inlet temperature, and cooling costs. Therefore, the workload scheduling strategy does not fully reflect the impact factors on the overall energy consumption of data centers.

Furthermore, prior studies demonstrated that workload consolidation is an effective approach to increasing resource utilization and lower power consumption [21] [22]. A growing number of workload consolidation-based techniques were proposed to reduce the power consumption of data centers. For example, Lee *et al.* [23] introduced a temperature prediction method for dynamic voltage and frequency scaling. Unfortunately, the above thermal management schemes overlook node failures incurred by increased node temperatures. A report by the Uptime Institute shows that when the node temperature exceeds 21°C, facility failures will double for every 10°C increase [16]. Unlike the aforementioned thermal-aware schemes, our HGSA takes into account both thermal awareness and failure factors to optimize lightweight thermal predictors.

Our HGSA is distinctly different from the aforementioned energy-saving schemes in two aspects. First, the above schemes merely consider workload consolidation, whereas our HGSA relying on air temperature distribution considers not only operational power consumption but also cooling costs and computing power consumption. Second, unlike the above-mentioned workload distribution policies that fail to consider the negative impact of system failure, our HGSA integrates the failure factor when making workload distribution decisions.

TABLE 1 summarizes the major technological discrepancies among HGSA and the existing schemes reported in the literature. We compare HGSA against the alternative schemes from five perspectives, including computing nodes modeling, thermal awareness, node failure mode, workload distribution modeling, and cooling cost consideration.

3 Modeling Data Centers

In this section, we propose a holistic data center model that seamlessly integrates both thermal awareness and node failures. Our data center model plotted in Fig. 1 consists of four components, namely, a workload model, a distributing system, a computing model, and a Computer Room Air Conditioner (CRAC) model.

Now let us first present the rationale behind our holistic model. In a data center, the workloads comprised of multiple user requests are submitted to the distribution system (see Fig. 1). Governed by a distribution strategy, the system dispatches the workloads across different nodes. Operations of computing nodes emit enormous heat inciting heat recirculation, which pushes up the temperature of computing nodes to a high level. If the inlet temperature of a node exceeds a certain safety threshold (i.e., redline temperature), the reliability and lifetime of the node will be adversely affected. To blow away heat by cool air, CRAC system is normally employed in a data center to expel hot air and supply cold air to outlets of computing nodes [13]. Due to the discrepancies in factors such as data blocks to be processed, locations of computing nodes, and network latency, the distribution of hot air in a data center becomes uneven, which sparks uneven heat recirculation that increases cooling costs. Therefore, heat recirculation effects represented by a cross-interference matrix are incorporated into our model, which characterizes the peak inlet temperature of computing nodes. As such, proper workload distribution can be performed in a data center to alleviate the impact of heat recirculation, thereby minimizing inlet temperatures and reducing cooling costs [10].

3.1 Workload Model

To reflect the impact of workloads on the power consumption of data centers, we mathematically model the workloads in a data center. We assume that a data center consists of N chassis; each of which houses M computing nodes; each computing node has m processors. Assuming there are $\tilde{c}_i$ workloads distributed on computing node i. In order to save energy, we render certain that all nodes in the data center can process the workloads distributed to them within the shortest time. Let workload represent the smallest unit of work handled by a node processor, and each processor only processes one workload at a time; $\tilde{C}_{total}$ denotes the total number of workloads in a data center; Pm_i denotes the ith processor in the Mth node, where $i \in [1, m]$. Now we have the total number of processors expressed as $2 * N * M = \{P1_1, P1_2, P1_m, P2_1, P2_2, P2_2...Pn_1, Pn_2, Pn_m\}$, where $n = M * N$ is the number of overall computing nodes of data centers. Thus, the workloads undertaken by a single processor is expressed as $Workload = \tilde{C}_{total}/2(N * M)$. It is worth noting that throughout this manuscript, we use the terms *a computing node* to represent *a server*. Fig. 2 depicts node locations in a data center.

If different amounts of workloads are allocated to the computing nodes in a chassis, the processing time incurred by the computing nodes with the same computing capability may be different. The processing time on computing node i can be expressed as:

$$t_i = [\frac{\tilde{c}_i}{m}], i = 1, 2 \cdots \tag{1}$$

where i denotes the node number, and m is the number of the processors in nodes i. $\tilde{c}_i$ denotes the number of workloads distributed on chassis i. Thus, we have a processing-time matrix for a workload distribution as:

$$\vec{t_i} = [t_1, t_2, \cdots, t_{\max([\frac{\tilde{c}_i}{m}])}], i = 1, 2 \cdots N \tag{2}$$

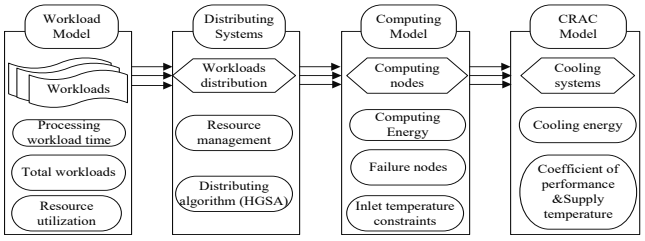


Fig. 1: Overall models of data centers for workloads distribution

3.2 Computing Cost Model

A data center consists of computing nodes, a power system, a CRAC system, etc. Among these components, computing nodes and CRAC are two dominating factors in power consumption, consuming more than 95% of the total power consumption [7].

To gauge the power consumption required to operate a data center, we propose a power consumption model as a sum of computing cost and cooling cost. We quantitatively model the energy cost caused by the computing node. Let us denote the power consumption of a single node in the idle state as *a*; the power consumption of a node in the active state is expressed as b; the power consumption of I/O and basic components are denoted as c. Recall (see Section 3.1) that each chassis has M nodes, we derive the following three expressions.

(a) The power consumption of a chassis with all its nodes running in the idle state is expressed as $P_{idle} = a * M + c$.

(b) The power consumption of a chassis with all its nodes running in the active state is expressed as $P_{run} = b * M + c$.

(c) The power consumption for a computing node to transition from the idle state to the running/active state is defined as $P_{u\sin g} = b - a$, which represents the startup power consumption of an idle computing node. When $\tilde{c}_i$ workloads are allocated to the computing nodes i, its power consumption is $P_{node} = a + \tilde{c}_i(b - a)$. Thus, the total power consumption of all computing node in a data center can be obtained as:

$$P_{IT} = \sum_{i=1}^{N} P_{node}^i \tag{3}$$

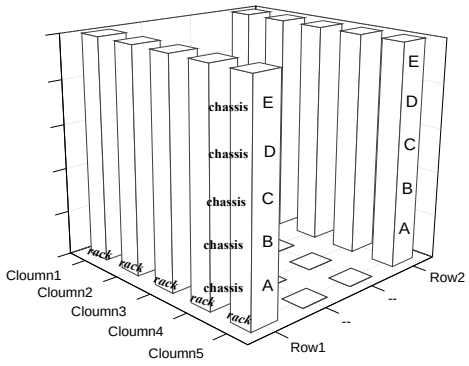


Fig. 2: A two-row data center, marked from bottom to top as A, B, C, D and E

3.3 CRAC Model and Total Power Cost of Data Centers

To characterize the cooling cost, we present the second part of our power consumption model. The increase of heat recirculation gives birth to the rising inlet temperature of computing nodes. To keep the inlet temperatures of computing nodes below a threshold (a.k.a, redline temperature), a CRAC is used for generating cool air to blow away heat. Unfortunately, the power consumption caused by CRAC can be as much as half of the total power consumption of a data center [8]. Therefore, we judiciously integrate CRAC impacts into our holistic model. The supply temperature $T_{\sup}$ of CRAC has a significant impact on cooling system efficiency. Typically, cold air generated by the cooling system flows into each rack through the inlet of each rack. Next, it blows away the generated heat and transforms from cold air to hot air. Then, the hot air is exhausted through ceiling return vents. The efficiency of heat recirculation is quantified by a Coefficient of Performance (CoP). CoP - a ratio of useful heating or cooling provided to work required - is defined as [4]:

$$CoP(T_{\sup}) = 0.0068T_{\sup}^2 + 0.0008T_{\sup} + 0.4580 \tag{4}$$

The heat recirculation and power consumption of computing nodes lead to an increase in its inlet temperature. Hence, we adjust the raised inlet temperature by using the temperature supplied by the CRACs, and the inlet

temperature cannot exceed the maximum allowed operational temperature (a.k.a, redline temperature) specified by the device manufacturer. Supply temperature T_{sup} affects the efficiency of the CRACs. The efficiency of CRACs is quantified by the CoP. Eq.(4) exhibits how the CoP increases with higher supply temperatures for the CRACs at the HP Labs Utility Data Center. The CoP is used to calculate the power consumption of the CRACs by using the following equation:

$$P_{AC} = \frac{P_{IT}}{CoP(T_{\text{sup}})} \tag{5}$$

As shown in Eq. (5), the higher the supply temperature, the less power the CRACs consume. However, the CRACs must supply more cold air to remove the heat generated by the computing nodes. Therefore, we can minimize P_{AC} by maximizing the supplied temperature T_{sup} while satisfying the constraints of the redline temperature. Therefore, we can derive the power consumption cost P_{Total} from this section and Section 3.2 as the sum of computing cost P_{IT} and cooling cost P_{AC}:

$$P_{Total} = P_{IT} + P_{AC} = \left(1 + \tfrac{1}{CoP}\right) P_{IT} \tag{6}$$

3.4 Inlet Temperature Model

Since the inlet temperature of a node is affected by the combination of cool air supplied by CRAC and heat recirculation effect from other computing nodes [9], we model the inlet temperature of a node as:

$$T_{in} = T_{\text{sup}} + [(K - A^T K)^{-1} - K^{-1}] * P_{IT} \tag{7}$$

where K denotes thermodynamic constants, expressing as a diagonal matrix $K_{n\times n} = diag(K_1, K_2, \cdots K_n)$. As a matter of convenience, we let $D = [(K - A^T K)^{-1} - K^{-1}]$. Thus, Eq. (7) is then re-formulated as:

$$T_{in} = T_{\text{sup}} + DP_{IT} = T_{\text{sup}} + DP_{idle} + DP_{u\sin g}C \tag{8}$$

For the above Eqs. (7) and (8), according to the law of energy conservation, the amount of heat by an airflow per unit time is $Q = \rho f c_p T$, where ρ is the air density; f is the airflow rate; c_ρ is the specific heat of air; T is the air temperature. Considering that the power drawn by a computing node is dissipated as heat, the steady-state relationship between power consumption of a node and the inlet/outlet temperature can be abstracted as $P^i_{IT} = \rho f_i c_p (T^i_{out} - T^i_{in})$, where $T^i_{out} = T^i_{in} + K_i P_i$, and $K_i = \rho f_i c_p$. Normally, a cross-interference coefficient matrix $A_{n\times n} = [\alpha_{ij}]_{N\times M}$ denotes how much of its outlet heat each computing node contributes to the inlet of every other node. Thus, the vector of inlet temperatures T_{in} can be expressed as Eqs. (7) and (8).

3.5 Failure Model of Computing Nodes

Existing data center thermal managements fall short in taking into account computing node failures. To bridge this gap, we propose a novel node failure model facilitating thermal-aware workload distribution.

We model heat recirculation with a cross-interference matrix. Fig. 3 depicts the cross interference of the heat recirculation effects among several neighboring nodes. The inlet temperature of $node_i$ is affected by CRAC temperature T_{sup} as well as its outlet temperature and heat re-circulation effects from other nodes. For this reason, we use matrix $A = [a_{ij}]_{N\times M}$ to represent the cross-interference of all the nodes in a data center [9]. Element a_{ij} in matrix A represents the fraction contribution of heat from node $node_i$'s outlet to $node_j$'s inlet temperature. The outlet temperature of $node_i$ impacts the inlet temperatures of itself (a_{ii}) as well as its neighboring node $node_j$ (a_{ij}). Thus, we have

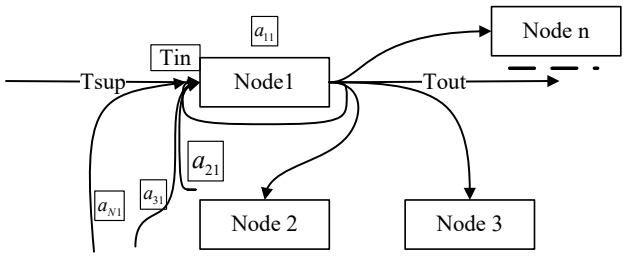


Fig. 3: data centers thermal-aware model

$$K(T_{out} - T_{\text{sup}}) = A' K(T_{out} - T_{\text{sup}}) + P_{IT} \tag{9}$$

where A' is the transpose of matrix A; T_{out} represents the output temperature vector of all nodes and $T_{out} = [T^1_{out}, T^2_{out}, \cdots T^n_{out}]$; T_{sup} is the temperature of the cold air supplied by the CRAC; P_{IT} denotes computing cost (see Section 3.2). Besides, K in Eq. (9) is a diagonal matrix, and $K_i = \rho f_i C_p$. To obtain matrix A, we set an initial reference for Eq. (9) and derive [3]:

$$K(T^{ref}_{out} - T_{\text{sup}}) = A' K(T_{out} - T_{\text{sup}}) + P^{ref}_{IT} \tag{10}$$

where P^{ref}_{IT} denotes the initial reference of computing cost; T^{ref}_{out} is the initial reference of the outlet temperature of all nodes. Based on Eqs. (9) and (10), we can now calculate the transpose of matrix A as:

$$A' = 1 - (P_{IT} - P^{ref}_{IT})(T_{out} - T^{ref}_{out})^{-1} K^{-1} \tag{11}$$

where P_{IT} is computing cost represented by a vector of power consumption of nodes and $P_{IT} = [P^1_{node}, P^2_{node}, \cdots P^n_{node}]$, $k = 1, 2, \cdots, n$; T_{out} is the vector of outlet temperatures of nodes and $T_{out} = [T^1_{out}, T^2_{out}, \cdots T^n_{out}]$.

Now we integrate failures into our thermal model [2]. Suppose node j fails, then we have $P^j_{node} = 0$ and $T^j_{out} = 0$ according to Eq. (9). In other words, faulty node j contributes no thermal recirculation, and it does not process any workloads. According to the above analysis, the change of the cross-interence coefficient matrix A' is expressed as $\alpha_{ij} = 0, i = 1, \cdots, n$ and $\alpha_{ji} = 0, i = 1, \cdots, n$. For example, we assume that a data center contains four computing nodes, which are $node_1, node_2, node_3, node_4$ failure. Therefore, the cross-interference coefficient can be expressed by A as

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix}.$$

If $node_2$ is a failure node, $node_2$ will not participate in workload distribution and generate heat. Therefore, $node_2$

will not be affected by heat recirculation with other computing nodes by the time $node_2$ fails, we have $\alpha_{i2} = 0, i = 1, \cdots, 4$ and $\alpha_{2i} = 0, i = 1, \cdots, 4$. The cross-interference coefficient matrix $A_{node_2 fails}$ can be represented as

$$A_{node_2 fails} = \begin{bmatrix} \alpha_{11} & 0 & \alpha_{13} & \alpha_{14} \\ 0 & 0 & 0 & 0 \\ \alpha_{31} & 0 & \alpha_{33} & \alpha_{34} \\ \alpha_{41} & 0 & \alpha_{43} & \alpha_{44} \end{bmatrix}.$$

In addition, the above analysis can be used to determine the cross-interference coefficient matrix with multiple failure nodes. For example, we assume $node_2, node_3$, the related cross-interference coefficient matrix $A_{node_{2,3} fails}$ will be

$$A_{node_{2,3} fails} = \begin{bmatrix} \alpha_{11} & 0 & 0 & \alpha_{14} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \alpha_{41} & 0 & 0 & \alpha_{44} \end{bmatrix}.$$

To model faulty nodes, we define matrix X for faulty nodes as $X = MZ$, where M denotes an unit diagonal matrix, $M = diag(x_1, \cdots x_n), x_i = 1, i \subset \{1, \cdots, n\}$; $Z = [z_1, z_2, \cdots, z_n]$ is a 1xn vector indicating faulty nodes' IDs; for node k, if k fails, z_k = 0; otherwise, if node k is active, z_k = 1.

We aim to construct a model guiding us to thermal-efficiently redistribute workloads by taking into consideration node failures. For this reason, we define a thermal-aware failure-based objective function for workload redistribution. Given N chassis and $n = N \times M$ nodes, we express the objective function for k failures as

$$\begin{aligned} &Minimize(\max\{T_{in}^i\}) \\ &st : \tilde{C}_{total} - \sum_{j=1}^{n} \tilde{c}_j = 0 \end{aligned} \tag{12}$$

$$T_{in} = T_{\sup} + X_k DX_k P_{u \sin g} + X_k DX_k CP_{idle}, k = (1, \cdots, n)$$
$$M \geq \tilde{c}_j \geq 0, j = (1, \cdots, n-k),\ u \leq (1 - k/n) * 100\%$$
$$(k+1)M \geq \tilde{c}_j \geq 0, j = (1, \cdots, n-k), u \leq (1 - k/n) * 100\%$$

where u denotes the utilization of a data center before nodes fail; T_{in}^i represents the inlet temperature of node i; C_{total} is the total workloads of all nodes, whereas $\tilde{c}_j$ is the distributed workload on chassis j, and $\tilde{C}_{total} = \sum_{j=1}^{l} \tilde{c}_j (l < n)$. Since the workload distribution problem is a NP-hard problem, it is judicious to make use of a heuristic algorithm to minimize $(\max\{T_{in}^i\})$, in other words, minimizing $T_{in} = T_{\sup} + X_k DX_k P_{u \sin g} + X_k DX_k \tilde{C}_{total} P_{idle}$, where $T_{\sup}$, $X_k DX_k$,$P_{u \sin g}$, and P_{idle} are given constants.

4 THE WORKLOAD DISTRIBUTING ALGORITHM

Recognizing that the workload distribution problem is an NP-hard problem, we leverage a heuristic optimization algorithm to find an approximate optimal solution. To this end, we propose *HGSA* - a hybrid algorithm based on *GA* and *SA* - to allocate workloads to computing nodes in a thermal-friendly manner. HGSA strives to reduce the total power consumption of data centers by minimizing the inlet temperatures of computing nodes. Designing such a hybrid algorithm is rational because GA exhibits stunning search capability in solving NP-hard problems while suffering from the weaknesses of local search [24]. On the other hand, simulated annealing algorithms (SA) featuring superb global search ability is an ideal choice to overcome the weakness of local search in GA [25].

4.1 HGSA Design

Guided by our failure-based thermal model, we design HGSA to optimize workload distribution in a way to conserve the total power consumption of data centers. The rationale behind the HGSA design is two-fold. First, HGSA adopts the strength of GA to quickly converge to an approximate optimal solution. Second, HGSA deploys SA to offset GA's weakness - being prone to get stuck in local optima. GA is commonly used to obtain approximate optimal solutions in solving workload distribution problems [9]. Through successive iterations, GA converges, terminates, and outputs an approximate optimal solution (i.e., an approximate optimal workload allocation vector). In each iteration, GA generates a workload distribution vector or *chromosome* as an input for its next iteration. A chromosome is comprised of multiple *genes*, each of which resembles a workload condition on a node. To speed up convergence, chromosome *crossovers* are performed by GA to obtain improved chromosomes in its iterations. A crossover operation is to cross the genes of two chromosomes at random positions with a certain probability. Unfortunately, random gene positions caused by chromosomes will give birth to a change in the total number of genes on crossover chromosomes. In other words, the total number of workloads distributed on all computing nodes will be changed, resulting in an excessive amount of computational overhead. To address this issue, HGSA replaces crossover operations with *reversal* operations. The principle of reversal operations is to only change the positions of the genes in a chromosome without changing the total number of genes in the chromosome.

Another inborn problem of the original GA is that amid searching for an approximate optimal workload-distribution vector, GA tends to get stuck in local optima. To improve the global searching ability of the original GA by jumping out of local optima, we combine the design principles of SA into our HGSA to alleviate GA's local optima issue. Simulated annealing (SA) algorithms were first introduced by Metropolis *et al.* [26]. The idea of SA is to simulate an annealing process in which temperature changes from high to low, and the energy gradually stabilizes to the lowest state; finally, a system reaches an equilibrium state. SA is adroit at approximate global optimization catering to large search space. By mixing the large-space-searching merit of SA with our HGSA, we make up for the shortcoming of GA.

In short, to boost the searching ability while speeding up convergence, our hybrid HGSA removes crossover operations, expands the searching scope, and avoids falling into local optima.

4.2 Workload Distribution Anchored on the Failure Model

If a computing node j fails, the distribution scheme $\vec{c}_{perf}$ is then used to obtain the number of workloads $\vec{c}_{perf}^{j}$

assigned to the failed node j. A rollback-recovery technique is exploited to periodically save the process state of the workloads to the storage of the computing node j (for example, the breakpoint when failures happen), thus avoiding restarting the current workload from the beginning on the failed computing node j. The storage system is assumed to be failure-free [17]. The proposed HGSA will select some candidate running computing nodes, and migrate the workloads $\vec{c}^{j}_{perf}$ from the failed computing node j to the selected nodes. Additionally, the process state of the workloads is also moved to the selected computing nodes. When the migration of workloads and process state is done, the migrated workload will restart from the breakpoint in terms of the process state.

Now we present HGSA's process of redistributing workloads on faulty nodes through the following three steps.

Step 1: Given z number of faulty nodes out of a total of n nodes, we denote the faulty nodes as $\{k_1, k_2, \cdots, k_z\}$. Since we leverage HGSA to output an optimal load distribution scheme $\vec{c}_{perf}$, we identify the corresponding node numbers of the failure nodes (i.e., $\{k_1, k_2, \cdots, k_z\}$) and obtain distributed workloads on each of the faulty node (i.e., $\{\vec{c}^{k_1}_{perf}, \vec{c}^{k_2}_{perf}, \cdots, \vec{c}^{k_z}_{perf}\}$). Thus, the total load distributed on all the faulty nodes is defined as $\vec{c}^{total}_{perf} = \vec{c}^{k_1}_{perf} + \vec{c}^{k_2}_{perf} + \cdots + \vec{c}^{k_z}_{perf}$.

Step 2: Because the faulty nodes no longer handle the assigned workloads, the total workloads allocated ($\vec{c}^{total}_{perf}$) on all faulty nodes must be redistributed to the remaining active nodes. To do so, we must first acquire a $n \times n$ heat distribution matrix W_z based on Eq. (7) and its derivations. Thus, we have

$$W_z = X_z[(K - A_z^T K)^{-1} - K^{-1}]X_z \tag{13}$$

Since there are z faulty nodes ($\{k_1, k_2, \cdots, k_z\}$), the faulty nodes' corresponding elements in matrix W_z are set to 0 and will be removed. Thereby, we obtain a $(n-z)\times(n-z)$ heat distribution matrix $W_z{}'$ for the remaining active nodes.

Step 3: Receiving the heat distribution matrix $W_z{}'$ for the remaining active nodes, the total workloads $\vec{c}^{total}_{perf}$ from all the faulty nodes will be redistributed to the remaining active nodes through our HGSA algorithm. Consequently, an optimal workload distribution decision for $\vec{c}^{total}_{perf}$ is generated by HGSA. Lastly, maximum execution time (see Section 3.1) determines whether the workloads redistribution plan meets the energy-saving requirements.

To sum up,the workload distribution plans are optimized under the guidance of our failure-based thermal model. In the same breath, our objective function Eq. (12) achieves the minimum inlet temperature of computing nodes, thereby reducing the power consumption of data centers.

4.3 Implementation Details

We articulate the details of HGSA aiming to minimize peak inlet temperature across all computing nodes to cut back total power consumption in data centers. In addition to the pseudo-code of HGSA outlined in Algorithms 1 and 2, TABLE 2 shows the symbols and the corresponding meanings used in HGSA.

Algorithm 1 HGSA

Input: $T_0, T_{end}, T, L, q, NIND, P_c, P_m, n, Head, T^{UB}, C_{total}, e, f$.
Output: workloads distribution scheme; Temperature of supplied air T_{sup}; Cooling cost of data centers.

Initial population;//generate random initial solutions;
$function(Solution, head, C_{total})$
While $T > T_{end}$ **do**
for $k \leftarrow 1$ to L **do**
$NewSolution()$; // evolutionary reversal operation and Mutation
$function(Solution, head, C_{total})$;//make the solutions meet the constraints of equation (13)
Metropolis principle (); //evaluate solutions
end for
$T \leftarrow q \times T$; // Cooling process
end While
return path

The HGSA algorithm (Algorithm 1) outlined above carries out the following four steps to minimize peak inlet temperature across all the computing nodes.

Step 1 Initial population: With the initialized parameters (see Table 2), HGSA yields a series of initial solutions (i.e., *chromosomes*, see Line 1). An initial solution (chromosome) contains *n genes*, where *n* denotes the total number of all computing nodes. Each gene is a random number from 0 to *m*, representing the amount of load allocated to a certain computing node (*n* nodes in total). Please note that $function()$ is employed to ensure that a random solution complies with the constraints stated in Eq. (12) (see Lines 2 and 6 in Algorithm 1).

The HGSA algorithm is a hybrid algorithm based on GA and SA. It can start from any solution and find an approximate optimal solution through continuous iterative optimization. Therefore, we use a random solution as the initial solution to lower the execution time of HGSA. This solution does not affect the approximate optimal solution found at the end.

TABLE 2: Symbols and corresponding meaning used in HGSA

Notation	Description
T_0	Initial temperature
T_{end}	Final temperature
T	Solid temperature
k	Counter
q	Cooling velocity
$NIND$	Population size
P_c	crossover probability
P_m	Mutation probability
n	Number of computing nodes in data centers
$Head$	The capacity of each node
$Solution$	workload distribution scheme

Step 2 Selection: The purpose of selection operations is to select the fittest chromosomes and pass their genes to the next generation. Based on the initial solutions, HGSA invokes Algorithm 2 to perform evolutionary reversal and mutation operations to generate new solutions (see Line 5 in Algorithm 1). The mutation strategy in HGSA is to randomly select evolutionary reversal positions on two chromosomes for recombination. More specifically, let two randomly generated positions on the two chromosomes be

r1 and *r2*, where both *r1* and *r2* $\in$ [1, n]. HGSA swaps the values of $r1$ and $r2$ to obtain a new solution (see Lines 1 - 14 in Algorithm 2).

Algorithm 2 Reverse operation and mutation operation

Input: $Solution, P_m, n, head, D,$
Output: NewSolution;
$Solution_{new} \leftarrow$ Reverse (solution, D)// Evolutionary reversal operation;
$function(solution, head, C_{total})$
While $rand > P_m$ **do**
Selecting a nonzero element $Solution_{new}[r1]$ randomly;
$R \leftarrow Solution_{new}[r1]$
$e \leftarrow$ random(1,R);//choosing a random number e between 1 and R
choose an element $solution_new[r2]$ randomly;
if $(r1 == r2)||(Solution_{new}[r2] + e) > head[r2]$
choose another element $Solution_{new}[r2]$ randomly;
end if
$Solution_{new}[r1] \leftarrow Solution_{new}[r1] - e;$
$Solution_{new}[r1] \leftarrow Solution_{new}[r1] + e;$
$NewSolution \leftarrow Solution_{new};$
end **While**

Step 3 Evaluation: Metropolis criterion is applied to evaluate generated solutions and determine whether or not a generated solution will be adopted as a new current solution (see Line 7 in Algorithm 1). Specifically, we set an initial solution at the maximum peak temperature to S; correspondingly, the peak temperature is $f(S)$. Next, we temporarily set the solution obtained from the reversal operation and mutation strategy in Step 2 as S'; therefore, the peak temperature value after reversal and mutation is $f(S')$. Then, we calculate the temperature difference as $\Delta T = f(S) - f(S')$. Given ΔT, we apply the following Metropolis criterion to assess the generated solution.

$$P = \begin{cases} 1 & \Delta T < 0 \\ \exp\left(-\frac{\Delta T}{T}\right) & \Delta T \geq 0 \end{cases} \quad (14)$$

If $\Delta T < 0$, accept S' as the new current solution, otherwise accept S as the new current solution with the probability $\exp(-\Delta T/T)$.

Step 4 Temperature Reduction: Given a cooling rate q, HGSA follows $T = q * T$ to repeatedly reduce peak inlet temperature. This temperature reduction process continues until T is lower than T_{end}; then, HGSA will cease the iteration process and output the current state as a final solution.

5 Evaluation

5.1 Experimental Setups

We conduct extensive simulation-based experiments on a computing node equipped with an Intel(R) Core Pentium(R) G3220 @3.00GHz CPU and a 4GB DDR3 RAM. The simulation platform used in the experiments is *matlab2016b*. We quantitatively evaluate the performance of HGSA by comparing it against five state-of-art algorithms, namely, UT, PSOGA, XINT-GA, SA, and PPVMP. We simulate a data center with a physical dimension of $9.6m \times 8.4m \times 3.6m$ (see Fig 2). This data center is housing ten industry-standard 42U racks, each of which contains fifty nodes contained in five chassis (labeled as A, B, C, D, and E from bottom to top). There is a total of 1000 processors in the data center; each processor is configured as a DELL PowerEdge 1855. The maximum number of workloads handled by a processor is 1000 (i.e., maximum $C_{total} = 1000$). The total power consumption is 2020W when all the nodes are in the idle state (i.e., the utilization of the data center is 0%). The energy overhead of transitioning from standby to active mode for a node is 50W [9]. A cold air with a flow rate of 8 m^3/s is provided through the CRAC outlet on the floor, blows away heat, and is exhausted as hot air through ceiling vents. Moreover, we use the *D* matrix provided by BLUESIM to evaluate the faulty node model. The thermal-aware matrix array is constructed by the BLUSIM tool powered by the CFD simulation [9].

5.2 Algorithms for Comparison

A variety of workload distribution strategies have been proposed to distribute workloads across computing nodes in data centers. Let us briefly introduce the basic ideas of five state-of-art schemes, namely, UT, XINT-GA, SA, PSOGA, and PPVMP.

UT (Uniform Task) [9]: Workloads are distributed evenly on all nodes. Consequently, the power consumption for each node will be: $\tilde{c}_i = {\tilde{C}_{total}}/{n}, \forall i \in [1, n]$. where n is the total number of nodes in a data center.

SA (Simulated Annealing) [2]: Simulated annealing is a stochastic global search optimization algorithm, which mimics the slow cooling of metals. The simulated annealing algorithm has attracted widespread attention due to its ability to jump out of local optima. We configure the initial temperature of SA to 10000°C, and its final temperature is set as 10^{-3} °C. Its maximum iteration at each temperature and the cooling rate is set to 150 and 0.9, respectively.

XINT-GA (Genetic Algorithm) [9]: XINT-GA allocates workloads across computing nodes to enhance CRAC cooling efficiency. GA is an iterative approach using a pool of genomes. Through a set of iterations, GA explores a solution space to reach a near-optimal solution. In each iteration, GA generates various solutions and then performs crossover and mutation operations to pick up one of them as the new solution. XINT-GA assigns workloads on computing nodes to minimize the maximum inlet temperature. In our experiments, we set the number of iterations of the compared GA and its probability of mutation to 300 and 0.05, respectively.

PSOGA (hybrid Particle Swarm Optimization and Genetic Algorithm) [27]: The hybrid PSO-GA algorithm efficiently assigns workloads to the resources to reduce the makespan and balance the workloads of the dependent workloads over the heterogeneous resource in cloud data centers. PSOGA crosses particles and self-mutates to search for an optimal solution. In our comparison experiments, the number of PSOGA's iterations is set to 300. Its population size and mutation rate are set to 100 and 0.05, respectively. Crossover operations belong to a single point. The acceleration coefficients C_1, C_2 are 1 and 1.1, respectively.

PPVMP(Power-aware and Performance-guaranteed Virtual Machine Placement) [14]: The PPVMP algorithm caters

to the problem of high PM power consumption and VM performance degradation. The PM power consumption model and VM performance model formulate VMP as a bi-objective optimization problem. The PPVMP extends the ant colony optimization to solve this problem. In the simulations, pheromone importance factor and heuristic informance importance factor are 1 ($\alpha = 1$, $\beta = 1$); local pheromone evaporating parameter and global pheromone evaporting parameter are 0.1 ($\rho_l = 0.1$, $\rho_g = 0.1$), fix parameter is 0.9 ($q_0 = 0.9$); the maximum iteration time is 300 ($N_{iter}^{\max} = 300$).

5.3 Selection and Analysis of HGSA Algorithm Parameters

We start our experiments by investigating the impacts of HGSA's parameters on its performance. Algorithm 1 suggests that the parameters affecting HGSA include cooling velocity q, crossover rate P_c, mutation probability P_m, initial temperature T_0, end temperature T_{end}, chain length L, the amount of initial population $NIND$, and the number of workloads C_{total}. Recall that (see Section 4.1) to avoid changing the total number of workloads distributed on computing nodes, HGSA - a hybrid of GA and SA - performs evolutionary reversal operations instead of crossover operations. For fair comparisons, the initial temperature T_0, end temperature T_{end}, chain length L, initial population $NIND$, and the number of workloads C_{Total} in HGSA are the same as those in the original GA and SA algorithms. More specifically, we have $T_0 = 1000$°C, $T_{end} = (10^{-3})$°C , $L = 150$, $NIND = 100$, and $C_{Total} = 1000$, where L is the number of iterations per workload T. Since the above-mentioned parameters remain unchanged, we pay heed to the impacts of cooling rate q and mutation probability P_m on HGSA's performance. We examine the effects of cooling rate and mutation probability on the maximum supplied temperature T_{sup}. Recognizing that the cooling rate and the probability of mutation are normally set to 0.8-0.99 [2] and 0.0001-0.1 [28], we let cooling rate q be 0.9 and 0.95 in our HGSA; besides, we set the mutation probability P_m to 0.05 and 0.1, respectively. The threshold temperature T^{UB} of computing nodes is 25°C. Because the focus of this group of experiments is the cooling rate q and mutation probability P_m, node failures are set to non-existent.

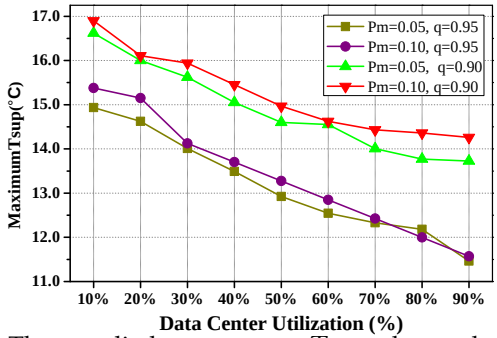


Fig. 4: The supplied temperature T_{sup} value under different parameters of cooling rate q and mutation probability P_m

Fig. 4 unveils the maximum supplied temperature T_{sup} as a function of cooling rate q and mutation probability P_m. We can observe from Fig. 4 that regardless of the values of cooling rate and mutation probability, the maximum supplied temperature under all the four configurations decreases as the data center utilization increases. When the cooling rate and mutation probability are set to 0.9 and 0.1, the supplied temperature T_{sup} has the highest value among the four configurations. In other words, the cooling cost is the lowest under this configuration. Thus, we adopt in this study a configuration where cooling rate q and mutation probability P_m are 0.9 and 0.1, respectively. Moreover, Fig.4 reveals that cooling rate has a higher impact on the supplied temperature T_{sup} than mutation probability. For example, the maximum supplied temperature T_{sup} is 16.8971°C when the data center utilization is set to 10%, the cooling rate q is set to 0.9, and the mutation probability P_m is set to 0.1. Under this configuration, a decrease of 0.05 in the mutation probability P_m causes a decline of 0.2796°C in the maximum supplied temperature T_{sup} (0.2796°C = 16.8971°C - 16.6175°C). In contrast, a surge of 0.05 (q is changed to 0.9 from 0.95) in the cooling rate leads to a reduction of 1.5202°C in the maximum supplied temperature T_{sup} (1.5202°C = 16.8971°C - 15.3769°C).

5.4 Correlations Between Failure Nodes

We are now in a position to characterize the temporal and spacial distributions of failures.

Spacial Distribution: The placement of nodes in a data center follow the Zipf distribution [15] [16], which can be expressed as:

$$B = f(r) * r^{\alpha} \tag{15}$$

where B is a constant; α is a skewness parameter. The lower the α, the more evenly the data nodes are placed. For example, when α is set to 0.01, nodes in a data center tend to be evenly distributed. However, when α increases to 0.09, the node distribution is likely to be skewed. In other words, node failures occur intensively in certain areas. Thus, it is prudent to reduce the number of node failures by tuning the parameters of the Zipf distribution of the node placement.

Temporal Distribution: Computing nodes in a data center temporally obey the *Weibull* distribution [15] [16]. We propose a system failure rate function, aiming to characterize occurrences of node failures concerning time. Assuming that node failures occur in the period between t and $t + \Delta t$, we express the failure rate in this period as

$$C(t) = \lim_{\Delta t \to 0} \left\{ \frac{p(t < T < t + \Delta t) | T > t}{\Delta t} \right\} = \frac{pdf(t)}{1 - prof(t)} \tag{16}$$

$$prof(t) = \left[{shape}/{scale} \right] \times \left[\frac{t}{scale} \right] \times e^{-\left[\frac{t}{shape} \right]^{shape}} \tag{17}$$

where $pdf(t)$ denotes a probability-density function, $prof(t)$ is a probability function. Then, the system failure rate that obeys the $weibull(scale, shape)$ distribution is calculated as:

$$C(t) = \frac{pdf(t)}{1 - prof(t)} = \left[\frac{shape}{scale} \right] \times \left[\frac{t}{scale} \right]^{shape-1} \tag{18}$$

Given Eq. (18), we analyze the reliability of computing nodes. Let us consider two faulty nodes housed in $chassis(K)$ and $chassis(L)$, where the next failure time

intervals are $uptime(K)$ and $uptime(L)$, respectively. Using Eq. (17), we calculate the failure rates of $chassis(K)$ and $chassis(L)$ as $CK = C(uptime(K))$ and $CL = C(uptime(LK))$. If $uptime(K) > uptime(L)$ and $shape < 1$, then we obtain $CK < CL$. Thus, $chassis(K)$ is more reliable than $chassis(L)$. The analyses reveal that (1) a node that has just failed has a high likelihood of failure; once the faulty node resumes work, it will become more stable. (2) It is mandatory to periodically perform reliability tests on recently failed nodes. More times than not, computing nodes in data centers obey failure rules based on temporal or special distribution. The correlation rule among failure nodes is mainly used to evaluate the reliability of each node. Our HGSA delineated in Section 4 redistributes workloads to select active nodes that exhibit high reliability. When a node fails, the node that has just failed is more likely to fail again. Therefore, it is necessary to shut down the failed nodes, which generates a significant impact on the heat recirculation among nodes. Our HGSA strategy can reduce the impact of heat recirculation through reasonable workloads distribution, thereby achieving energy saving in a data center.

5.5 Evaluating the Efficiency of CRAC with Failures

In this group of experiments, we evaluate the impacts of our HGSA on the CRAC efficiency in terms of minimum inlet temperature T_{in}^i, maximum supplied temperature T_{sup}, and cooling cost P_{AC}. We compare our HGSA with the five state-of-art schemes - UT, PSOGA, XINT-GA, PPVMP, and SA in four failure scenarios. The threshold temperature T^{UB} (redline temperature) of computing nodes is set as 25°C.

5.5.1 ***Scenario 1:*** *The Non-failure Case*

In the first scenario, the number of failures is zero for all the six compared schemes. We denote these six zero-failure workload distribution schemes as NF-PSOGA, NF-XINT-GA, NF-UT, NF-SA, NF-PPVMP, and NF-HGSA. Fig. 5 shows the CRAC efficiency of the six zero-failure schemes with respect to data center utilization. Fig. 5(a) depicts the minimum inlet temperature T_{in}^i of NF-PSOGA, NF-XINT-GA, NF-UT, NF-SA, NF-PPVMP, and NF-HGSA as a function of the data center utilization. We keep the supplied temperature T_{sup} to 10°C in this experiment. Fig. 5(a) reveals that regardless of the workload distribution schemes, the minimum inlet temperature increases as the data center utilization grows. The scheme without the heat perception ability (NF-UT) leads to the highest inlet temperature, whereas our NF-HGSA obtains the lowest minimum inlet temperature T_{in}. Recall that the lower T_{in} a workload distribution algorithm yields, the higher CRAC efficiency. We conclude that a data center managed by NF-HGSA has the highest cooling efficiency among its peers. Fig. 5(b) depicts the maximum supplied temperature T_{sup} of NF-PSOGA, NF-XINT-GA, NF-UT, NF-SA, NF-PPVMP, and NF-HGSA as a function of the data center utilization. We configure the threshold temperature T^{UB} of computing nodes to 25°C. Fig. 5(b) indicates that the maximum supplied temperature of the six workload distribution schemes plummets as the data center utilization goes up. Regardless of the utilization rate of the data center, NF-HGSA always leads to the highest maximum supplied temperature T_{sup} compared with the other competitors. The findings confirm that with NF-HGSA in place, the CRAC consumes the lowest energy because a high supplied temperature helps to conserve the power consumption of CRAC.

Fig. 5(c) shows the cooling cost P_{AC} of the six workload distribution schemes under various data center utilization. The results plotted in Fig. 5(c) illustrate that our NF-HGSA is superior to the other schemes in terms of conserving cooling energy. In addition, as the data center utilization climbs to more than 10%, cooling-cost savings in NF-HGSA becomes more remarkable.

5.5.2 ***Scenario 2:*** *The Single-failure Case*

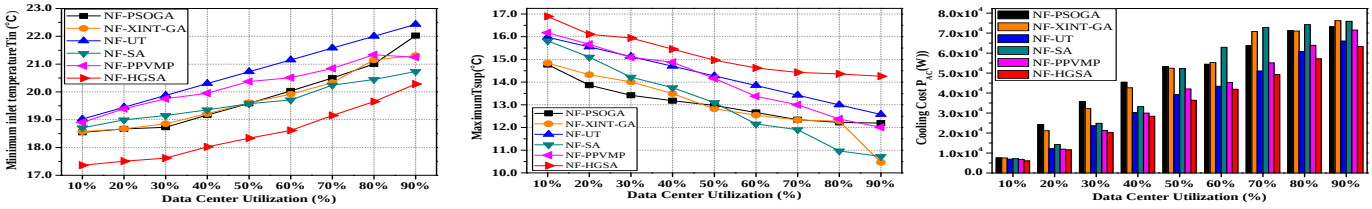
In the second scenario, we set the number of failures to one by randomly selecting a faulty node in the simulated data center. Similar to *Scenario 1*, we compare HGSA with the other six schemes in terms of the efficiency of a single-failure data center. We denote these six schemes in this single-node-failure scenario as F-PSOGA, F-XINT-GA, F-UT, F-SA, F-PPVMP, and F-HGSA, respectively.

Fig. 6(a) and (b) exhibit similar trends to those of Fig. 5(a) and (b). Regardless of the utilization rate, our F-HGSA outperforms F-XINT-GA, F-UT, F-SA, F-PPVMP, and F-PSOGA, harvesting the lowest minimum inlet temperature T_{in}^i and the highest maximum supplied temperature T_{sup}. Fig. 6(c) demonstrates that our F-HGSA has a clear edge over the F-PSOGA, F-XINT-GA, F-SA, F-PPVMP, and F-UT algorithms in terms of the cooling cost. Among these six algorithms, the performance of F-UT is close to our F-HGSA. This close performance is expected because F-UT distributes workloads evenly on all nodes, where one faulty node has an insignificant effect on overall performance.

5.5.3 ***Scenario 3:*** *The Multiple-failure Case*

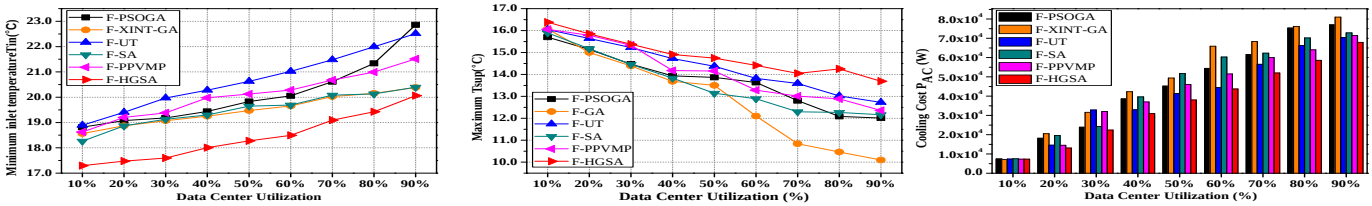
To investigate the impact of multiple failures on the CRAC efficiency, we set up in this scenario multiple faulty nodes in our simulated data center. We denote the six workload distribution schemes in the multiple-failure case as MF-HGSA, MF-PSOGA, MF-XINT-GA, MF-SA, MF-PPVMP, and MF-UT. In the multi-faulty node scenario here, we set the number of faulty nodes to 3. Fig. 7 depicts the CRAC efficiency of the six schemes under various data center utilization. Fig. 7(a) shows our MF-HGSA achieves the lowest minimum inlet temperature T_{in} among the six strategies. MF-HGSA's advantage is attributed by the fact that as the number of failures increases, our MF-HGSA is able to redistribute workloads of faulty nodes to the other normal active nodes that generate less heat. Moreover, Fig. 7(a) indicates that except for our MF-HGSA, multiple-node failures spur an increasing burden on the CRAC in the cases of the other workload distribution schemes. For example, the minimum inlet temperature of MF-PSOGA sharply goes up when the data center utilization exceeds 60%.

Fig. 7(b) reveals that our MF-HGSA exhibits the best performance in terms of maximum supplied temperature. More specifically, MF-HGSA obtains the highest maximum supplied temperature T_{sup} across all the data center utilization. The maximum supplied temperature of MF-HGSA decreases slower than those of the five alternatives when the data center utilization goes up. This effect becomes more



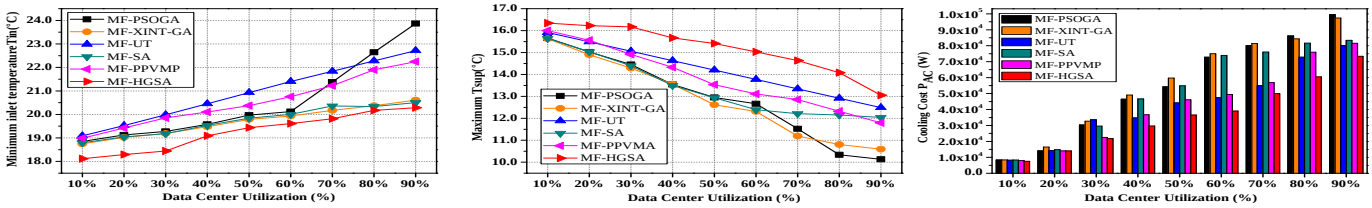
(a) Minimum T_{in} (b) Maximum T_{sup} (c) Cooling Cost P_{AC}

Fig. 5: Minimum inlet temperature T_{in}, Maximum T_{sup} and Cooling Cost P_{AC} of the different algorithms in **Scenario 1**



(a) Minimum T_{in} (b) Maximum T_{sup} (c) Cooling Cost P_{AC}

Fig. 6: Minimum lnlet temperature T_{in}, Maximum T_{sup} and Cooling Cost P_{AC} of the different algorithms in **Scenario 2**



(a) Minimum T_{in} (b) Maximum T_{sup} (c) Cooling Cost P_{AC}

Fig. 7: Minimum lnlet temperature T_{in}, Maximum T_{sup} and Cooling Cost P_{AC} of the different algorithms in **Scenario 3**

pronounced if the utilization exceeds 60%. Unlike the other five techniques, the maximum supplied temperature of HGSA in this multiple-failure scenario is nearly the same as that of HGSA in the single-failure counterpart. For example, the range of T_{sup} obtained by HGSA in the single-failure scenario is [13.6912°C, 16.3841°C], and the range in the multiple-failure scenario becomes to [13.0492°C, 16.3398°C]. In other words, our HGSA is less sensitive to node failures compared with the other five solutions. Fig. 7(c) shows the cooling cost caused by the six distribution schemes under various utilization. Fig. 7(c) demonstrates a similar trend as the one depicted in Fig. 6(c). That is, MF-HGSA outpaces the other five competitors in terms of conserving cooling energy. The other schemes' cooling costs reduced by our MF-HGSA become larger as the data center utilization grows. The reduced cooling cost reaches its peak when the data centers utilization is 90%. Our experimental results confirm that when the utilization is low, a node failure has a negligible impact on cooling performance. However, when the utilization is 30%, the cooling cost of a single failure and multiple failures is higher than that of a non-failure by an average of 1.18%, 13.03%. When the utilization is 40%, the cooling cost of single failure and multiple failures is higher than that of a non-failure by an average of 1.32%, 15.02%. This cost grows with increasing utilization. We also refer to a related study for this mechanism [16], where the impact of failures is a critical factor to the waste of resources and low energy efficiency.

5.5.4 ***Scenario 4:*** *The Impact of Failure Locations*

In this scenario, we examine the impact of failure locations on maximum supplied temperature T_{sup}. To this end, we investigate the failure location impact with respect to two cases. Fig. 8 depicts the maximum supplied temperature as a function of failure locations. Recall that our simulated data center embraces ten racks placed in *Row 1* and *Row 2*, where each rack has five chassis labeled as A, B, C, D, and E from bottom to top to house computing nodes (see Fig. 2). In Fig.8, $D1$ and $E1$ in *Row 1* denote node failures near the top of rack 1; $A1$ and $B1$ in *Row 1* denote node failures near the bottom of rack 1. Similarly, $D2$ and $E2$ in *Row 2* represent node failures near the top of rack 2; $A2$ and $B2$ in *Row 2* denote node failures near the bottom of rack 2.

Case 1: Failures occur at multiple locations in the same rack.

In this case, we compare the maximum supplied temperature T_{sup} where node failures occur near the top against that where node failures occur near the bottom in the same rack. For this reason, we compare T_{sup} where nodes $D1$ and $E1$ in Row 1 fail against the counterpart T_{sup} where nodes $A1$ and $B1$ in Row 1 fail (similarly, $D2$ and $E2$ in Row 2 vs. $A2$ and $B2$ in Row 2). Fig. 8 reveals that regardless of the utilization, the T_{sup} with faulty nodes $A1$ and $B1$ is greater than that with faulty nodes $D1$ and $E1$. Compared with node failures that occur near the bottom of racks, node failures near the top of racks lead to a lower maximum supply temperature consuming more energy in data centers.

Case 2: Failures occur at the same locations in different

racks.

In this case, we investigate the maximum supplied temperatures T_{sup} caused by node failures that occur at the same locations in different racks. For example, we compare T_{sup} caused by node failures near the top of rack 1 (i.e., $D1$ and $E1$ in Row 1) against that incurred by node failures near the top of rack 2 (i.e., $D2$ and $E2$ in Row 2) under various data center utilization. Fig. 8 unfolds that there is no obvious gap between these two curves. This finding indicates that node failures that occur at the same locations in different racks have similar impacts on the CRAC efficiency.

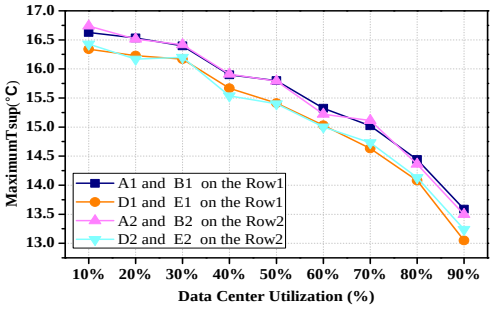


Fig. 8: The experiments of failure nodes residing in different positions in **Scenario 4**

5.6 Total Power Consumption and Execution Time

Now we compare HGSA with PSOGA, XINT-GA, UT, PPVMP, and SA in terms of power consumption and execution time under different failure scenarios.

Total Power Consumption Evaluation: Fig. 9 depicts the total power consumption of HGSA, PSOGA, XINT-GA, UT, PPVMP, and SA with respect to data center utilization in three failure scenarios. Compared with PSOGA, XINT-GA, UT, PPVMP, and SA, our HGSA reduces the total power consumption of a data center. More importantly, when the data center utilization becomes high, HGSA still maintains its energy-efficiency edge over the other five techniques. Such an energy-saving effect becomes more pronounced in the multiple-failure scenario (Scenario 3). For example, our HGSA cuts down the power consumption of XINT-GA by approximately 24% when the utilization is 90% in Scenario 3. Fig. 9 also shows that the total power consumption of UT is relatively high across all the utilization measures and the node failure scenario. The reason is that UT is a non-failure-aware load distributor, which has low energy efficiency. Moreover, we observe that the total power consumption of data centers with fewer faulty nodes is always greater than that with multiple faulty nodes. This result is expected because node failures give rise to a concentrated workload distribution to a small number of active nodes, which strengthens the heat recirculation effect, thereby increasing the total power consumption. In a nutshell, HGSA is more thermal-efficient than the other workload distribution algorithms in multiple-node-failure environments. First, the HGSA algorithm can optimize the cost of the cooling system by finding an approximate optimal solution. Second, HGSA can optimize the power consumption of computing nodes by controlling the number of nodes activated. Therefore, compared to other state-of-the-art methods, HGSA can reduce more total power consumption of data centers.

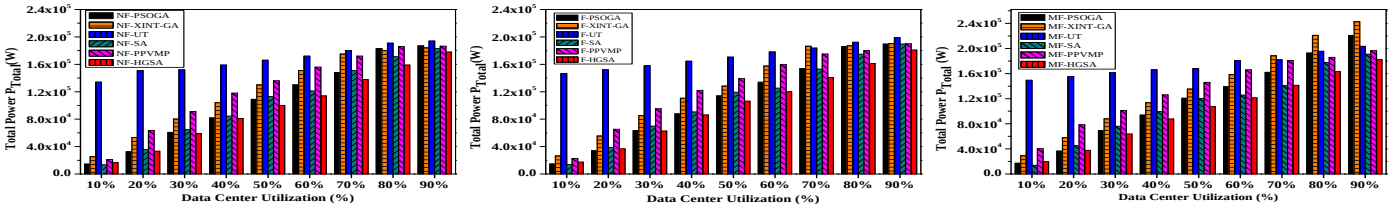
Execution Time Evaluation: Fig. 10 plots the execution time of these six schemes to obtain the minimum inlet air temperature. We conduct this group of experiments under various data center utilization in three failure scenarios. The execution time of UT is much shorter than those of PSOGA, XINT-GA, SA, PPVMP, and our HGSA. This is because these five schemes are iterative-based algorithms in nature. Fig. 10 also illustrates that the execution time of the five algorithms is insensitive to the data center utilization. HGSA's execution time spent in distributing load is higher than those of SA and UT; HGSA is faster than PSOGA, PPVMP, and XINT-GA. We conclude that the execution time of HGSA falls in a reasonable range. Moreover, Fig. 10(a), (b), and (c) indicate that although the number of node failures varies in scenarios 1, 2, and 3 (zero-failure, single-failure, and multiple-failure scenarios), the execution time of each algorithm does not significantly increase.

5.7 The Impact of Failure Rate

Schroeder and Gibson [15] investigated the failure rates of 22 high-performance computing (HPC) systems composed of 4750 computing nodes at Los Alamos National Laboratory (LANL). Their survey results show that average failure rates differ wildly across systems, ranging from 20-1000 failures per year, and that time between failures is modeled well by a Weibull distribution with decreasing hazard rate. Furthermore, their results indicate that the failure rate of computing nodes in a period is greatly related to the number of workloads or the utilization rate of the data center. Based on the above analysis, we evaluated the total power consumption of the data center by optimizing workload distribution under different failure rates by using different algorithms. Fig. 11 shows that the HGSA algorithm achieves the lowest total power consumption among the six schemes. It is observed that the total power consumption has a visible discrepancy with the node failure rate ranging between 10% and 50%. This is because heat recirculation has a great influence on the cooling power consumption of data centers when the failure rate is less than 50%, thereby affecting the total power consumption of data centers. This means that the failure of computing nodes is a critical factor to lower energy efficiency. However, if the failure rate varies between 60% and 90%, the total power consumption under different algorithms is significantly reduced. This is because all normal computing nodes are required to participate in workload processing, which makes the effects of heat recirculation more stable. Unfortunately, the total power consumption decreases rapidly as the failure rate increases due to the excessive failure rate of computing nodes. Additionally, our HGSA algorithm performs better in terms of reducing total power consumption under different node failure rates compared with UT, SA, XINT-GA, PSOGA, and PPVMP. This is because the HGSA algorithm enables the workloads to be distributed to computing nodes where the heat recirculation affects less compared with other schemes.

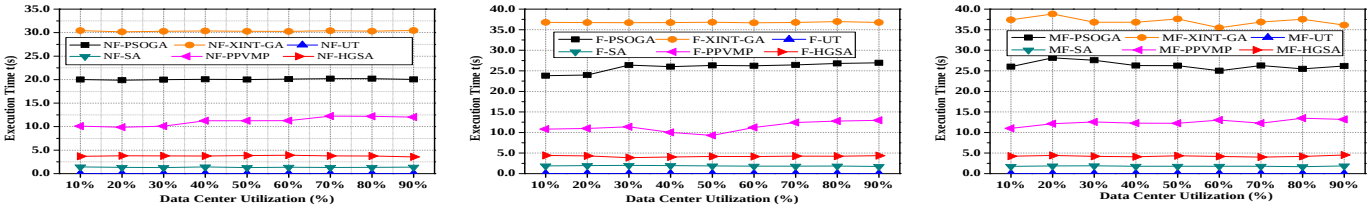
6 Conclusion

In this paper, we proposed a heuristic algorithm - HGSA - to optimize workload distribution driven by a holistic failure-aware model, with the purpose of boosting the thermal and



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 9: Total power consumption (i.e.,cooling and IT equipment cost) of data centers managed by the different algorithms in a) Scenario 1 (see Section 5.5.1), b)Scenario 2 (see Section 5.5.2), and c) Scenario 3 (see Section 5.5.3)



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Fig. 10: Execution time of the HGSA, PSOGA, XINT-GA, UT, PPVMP, and SA algorithms.

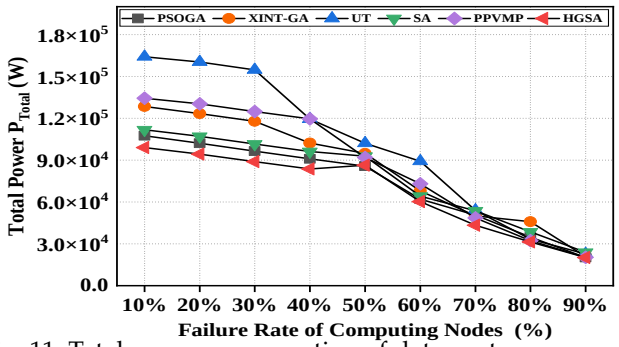


Fig. 11: Total power consumption of data centers managed by the different algorithms with different failure rates

energy efficiency of data centers. We introduced a novel model to characterize the energy efficiency of data centers by taking into account including workloads, computing and cooling costs, inlet temperature, and node failures. Moreover, we delineated temporal–spacial distributions of node failures in our model to precisely reflect the impact of node failures.

We carried out extensive experiments to quantitatively evaluate the performance of our proposed HGSA by comparing it with SA, XINT-GA, PSOGA, PPVMP, and UT. The experimental results unveil that HGSA outperforms the other five methods in terms of minimum inlet temperature, maximum supplied temperature, cooling cost, and total power. In addition, our experiments have also confirmed that node failures cause an increase in cooling cost and the overall power consumption of data centers. Meantime, the findings suggest that the location of faulty nodes affects the cooling temperature and power consumption of data centers. Specifically, the power consumption of a data center with faulty nodes near the top of racks is higher than that with faulty nodes near the bottom of racks.

As a future research direction, we plan to extend our research to secondary-storage-level failures. We will develop a workload redistribution strategy that facilitates data movement from faulty disk drives to functioning ones.

Acknowledgment

This work is sponsored by the National Natural Science Foundation of China under Grant No. 62072214, the Guangdong Basic and Applied Basic Research Foundation under Grant No. 2021B1515120048, the International Cooperation Project of Guangdong Province under Grant No. 2020A0505100040, and the Open Project Program of Wuhan National Laboratory for Optoelectronics No.2020WNLOKF006. The corresponding author of this paper is Yuhui Deng.

References

[1] R. Sahoo, M. Squillante, A. Sivasubramaniam, and Y. Zhang, "Failure data of a large-scale heterogeneous server environment," in *IEEE International Conference on Dependable Systems and Networks*, 2004, pp. 772–781.

[2] H. Feng, Y. Deng, and L. Yu, "Modeling the failures of power-aware data centers by leveraging heat recirculation," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 7, 2019.

[3] 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.

[4] 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*, 2005.

[5] T. Chalermarrewong, T. Achalakul, and S. C. W. See, "Failure prediction of data centers using time series and fault tree analysis," in *Proceeding of the 18th IEEE International Conference on Parallel and Distributed Systems*, 2012, pp. 794–799.

[6] 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, pp. 102 048–102 062, 2021.

[7] W. Lin, F. Shi, W. Wu, K. Li, and G. Wu, "A taxonomy and survey of power models and power modeling for cloud servers," *ACM Computing Surveys*, vol. 53, no. 5, pp. 1–41, 2018.

[8] 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, 2016.

[9] Q. Tang, 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.

[10] Y. Chen, M. I. Alghamdi, X. Qin, J. Zhang, M. Jiang, and M. Qiu, "Tern: A self-adjusting thermal model for dynamic resource provisioning in data centers," in *Proceeding of the 17th IEEE International Conference on High Performance Computing*, 2015, pp. 479–490.

[11] A. Rosa, L. Y. Chen, and W. Binder, "Failure analysis and prediction for big-data systems," *IEEE Transactions on Services Computing*, vol. 10, no. 6, pp. 984–998, 2017.

[12] T. Van Damme, C. De Persis, and P. Tesi, "Optimized thermal-aware job scheduling and control of data centers," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 2, pp. 760–771, 2019.

[13] 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. 6, pp. 1317–1331, 2018.

[14] H. Zhao, J. Wang, 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.

[15] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 4, pp. 337–350, 2010.

[16] P. Garraghan and I. S. M. et al., "An analysis of failure-related energy waste in a large-scale cloud environment," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 166–180, 2014.

[17] Y. Sharma, W. Si, D. Sun, and B. Javadi, "Failure-aware energy-efficient vm consolidation in cloud computing systems," *Future Generation Computer Systems*, vol. 94, pp. 620–633, 2019.

[18] G. Wang, L. Zhang, and W. Xu, "What can we learn from four years of data center hardware failures?" in *Proceedings of the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2017, pp. 25–36.

[19] A. Alquraan, H. Takruri, M. Alfatafta, and S. Al-Kiswany, "An analysis of network-partitioning failures in cloud systems," in *Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation*, 2018, p. 51–68.

[20] I. S. Moreno, P. Garraghan, P. Townend, and J. Xu, "Analysis, modeling and simulation of workload patterns in a large-scale utility cloud," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 208–221, 2014.

[21] Y. Zhou, S. Taneja, G. Dudeja, X. Qin, J. Zhang, M. Jiang, and M. I. Alghamdi, "Towards thermal-aware hadoop clusters," *Future Generation Computer Systems*, vol. 88, pp. 40–54, 2018.

[22] 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.

[23] J. S. Lee, K. Skadron, and S. W. Chung, "Predictive temperature-aware dvfs," *IEEE Transactions on Computers*, vol. 59, no. 1, pp. 127–133, 2010.

[24] M. Kumar, M. Husain, and et al., "Genetic algorithm: Review and application," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 451–456, 2010.

[25] J. Suarez, C. Millan, and M. Millan, "Improved modified simulated annealing algorithm for global optimization," *Contemporary Engineering Sciences*, vol. 11, no. 96, pp. 4789–4795, 2018.

[26] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[27] A. M. Manasrah, H. Ba Ali, and B. B. Gupta, "Workflow scheduling using hybrid ga-pso algorithm in cloud computing," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–16, 2018.

[28] L. Yang, Y. Deng, L. T. Yang, and R. Lin, "Reducing the cooling power of data centers by intelligently assigning tasks," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1667–1678, 2018.

Jie Li received the B.E. degree in automation from Wanjiang 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 currently pursuing the PHD degree with the Computer Science Department, Jinan University. His current research interests include data center architecture, cloud computing and data replica placement.

Yuhui Deng is a professor at the Computer Science Department of Jinan University. He worked as a research officer at Cranfield University in the United Kingdom from 2005 to 2008. He received his Ph.D. degree in computer science from Huazhong University of Science and Technology in 2004. His research interests cover information storage, cloud computing, green computing, computer architecture, performance evaluation, etc.

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. His research interests include energy-saving techniques, database systems, big data techniques and parallel computing.

Zhen Zhang received the BS and MS degrees in computer science from Jilin University, China, in 1999 and 2003, and the PhD degree from the College of Computer Science and Engineering, South China University of Technology, China, in 2011. He became a lecturer and an associate professor, in 2003 and 2012, respectively, in the department of computer science at Jinan University. His research interests include parallel and distributed processing and complex networks.

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. His research interests include High Performance Computing, Ubiquitous Computing, Modelling and Performance Engineering.

Xiao Qin received 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 wih the Department of Computer Science and Software Engineering, Auburn University. His research interests include parallel and distributed systems, storage systems, fault tolerance, real-time systems, and performance evaluation.