2015 IEEE 17th International Conference on High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), and 2015 IEEE 12th International Conf on Embedded Software and Systems (ICESS)

TERN: A Self-Adjusting Thermal Model for Dynamic Resource Provisioning in Data Centers

Yuanqi Chen*, Mohammed I. Alghamdi†, Xiao Qin*, Jifu Zhang§, Minghua Jiang¶ and Meikang Qiu‖
* Department of Computer Science and Software Engineering, Auburn University, Auburn, AL, 36849-5347 USA.
† Department of Computer Science, Al-Baha University, Al-Baha City, Kingdom of Saudi Arabia
§ School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China.
¶ College of Mathematics and Computer Science, Wuhan Textile University, Wuhan, China
‖ Department of Computer Science, Pace University, New York, NY 10038, USA
Email: *yzc0029@tigermail.auburn.edu, xqin@auburn.edu †mialmushilah@bu.edu.sa, §zjf@tyust.edu.cn, ¶jmh@wtu.edu.cn, ‖mqiu@pace.edu

***Abstract*—Dynamic resource provisioning becomes a practical approach to achieving high thermal and energy efficiency, improving scalability, and optimizing reliability for e-commercial applications running in modern data centers. In this paper, we propose a self-adjusting model called TERN to predict thermal behaviors of hardware resources for client sessions. Our TERN contains two major components: (1) a resource utilization model being responsible for estimating hardware usage based on the number of running client transactions, and (2) a thermal model that discovers correlation between resource utilization and their temperatures. TERN is conducive to predicting thermal trends of diverse workload conditions with a changing transaction mix. We apply the TPC-W benchmark to characterize the resource demands of each type of transactions. Then, we conduct a thermal profiling study of the benchmark with various transaction mixes. TERN judiciously adjusts the models to maintain prediction accuracy for dynamically changing request patterns. Experimental results show that TERN provides a simple yet powerful solution for resource provisioning in thermal-aware data centers where exist rapidly changing workload conditions.**

I. Introduction

Thermal management plays a vital role in modern data centers. Traditional thermal models are mainly focused on static and non-changing workloads. To address this issue, we propose in this paper a self-adjusting thermal model called TERN to predict thermal behaviors of hardware resources provided for client sessions. We characterize the resource usage and thermal behaviors for various transaction types. We show how to apply TERN to predict thermal trends for three real-world scenarios under changing workload conditions.

Our thermal model is motivated by the following four factors in the context of data centers:

- The growing importance of thermal management in modern data centers.
- The pressing needs of prediction models to guide dynamic resource provisioning.
- Fine granularity modeling for resource provisioning and thermal optimization.
- Self-adjustment for rapidly changing transaction mixes.

Data centers house computing clusters that consist of thousands or even ten thousands of servers. A current trend shows that computing and storage capacity demands of data centers exponentially increase due to big data explosion [1]. Gartner predicted that a big data center's hardware expense will take up 71 percent (almost $126.2 billion) of the worldwide data centers hardware expense [2]. Such a huge infrastructure brings a surge of supplying energy cost, which is in the order of millions of dollars (see also recent reports in [3]). Two major contributing factors of energy cost in data centers: energy consumed by data nodes (e.g., CPU, disk, memory, fans, network) and energy caused by cooling systems (e.g., *Computer Room Air Conditioners (CRACS)*). Cooling cost can be as high as half of the total energy costs depending on data centers' power usage effectiveness (PUE). There is a large body of studies on run-time reduction of cooling and energy cost through thermal-aware task dispatch and migration strategies [4][5].

It is not uncommon for a 1000-rack data center to consume 10MW total power per year [6]. A handful of studies have been focused on modeling energy efficiency of data centers. For example, Allalouf *et al.* introduced an energy model to estimate power consumption of storage cluster in data center [7]. Pan *et al.* simulated data centers using their developed energy models [8]. Increasing attention has been paid to the development of high performance and energy-efficient clusters in data centers [9]. Apart from these energy efficiency studies, there is a large body of thermal management techniques tailored for clusters in data centers.

There are two factors driving us to optimize thermal efficiency in data centers. First, cooling cost can be reduced by minimizing energy consumed by cooling infrastructures coupled with advanced thermal management techniques [10][11]. Second, maintaining low cooling cost for large-scale data centers with an increasing number of nodes becomes a grand challenge [12][13]. Bringing down energy consumption of cooling systems largely depends on thermal behaviors of data centers. Several approaches have been investigated to avoid high temperatures in data centers. For example, preventing server temperatures from moving above a red-line (i.e., *hot spots*) reduces hot air recirculated from outlets into inlets of nodes. Thermal management not only decreases cooling cost, but also improves server reliability. The thermal model developed in our study can be adopted for building thermal management modules. In addition, our model

978-1-4799-8937-9/15 $31.00 © 2015 IEEE
DOI 10.1109/HPCC-CSS-ICESS.2015.183

IEEE computer society

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

provides a guideline for resource provisioning of applications like web searching and e-commercial under rapidly changing workload conditions.

Internet services have become an indispensable part of people's everyday life. These services are expected to instantly respond to requests while guaranteeing performance requirements. Any delay in request response may cause thousands of dollar losing. Typical internet services tend to exhibit dynamical regular long-term (e.g., weekly or daily) variations as well as short-term fluctuations due to flash crowds (e.g., holiday season sales) [14]. There are numerous documented examples of Internet applications that faced an outage due to an unexpected overload. These outage problems reveal that well predicting patterns of services (e.g., its peak workload) and real-time monitoring are extremely important for dynamic resource provisioning to deal with unexpected overload. For instance, Amazon's website suffered a forty-minute downtime due to an overload during the popular holiday season in November 2000 [15]. Most internet services consist of an assortment of transactions; analyzing each type of transaction request in terms of resource demands will be of help for real-time resource provisioning. Profiling internet services at the request level not only offers fine-granularity resource allocations to requests, but also provides thermal information to improve energy efficiency of data centers.

It is challenging to construct a thermal model for dynamic resource provisioning in the context of internet applications in data centers because of the following three reasons.

- Each workload has its own resource-requirement characteristics (e.g., some are computing intensive whereas others are data-intensive).
- A wide range of hardware settings have various thermal behaviors.
- Integrating multiple models to predict thermal trend may worsen prediction errors.

In this study, we adopt the divide and conquer approach to seamlessly integrate two models into TERN to overcome the aforementioned challenges. The first model is a resource utilization model, which is responsible for building up correlation between hardware utilization and request patterns (e.g., arrival rate). The second one is a thermal model to predict thermal behaviors of processors and disks. We integrate these two models into TERN, which can adjust modeling parameters to achieve low prediction errors for dynamically changing request patterns.

The rest of paper is organized as follows. Section II summarizes the related works and the experimental testbed is described in Section III. In Section **??**, we delineate the design of TERN in a three-tier architecture. Then, we discuss the development details of TERN in Section V. Section VI shows how to make TERN adjust modeling parameters to minimize prediction errors. The models integrated in TERN are validated with the three workload mixes in Section VII. Conclusions and future work are outlined in Section VIII.

II. Related Work

A. Resource Management and Provisioning

There are a handful of studies focusing on resource management and provisioning. For instance, Doyle *et al.* proposed a model-based approach to utility resource management for coordinated provisioning of memory and storage resources [16]. This approach enables systems to achieve efficient resource management, including differentiated service quality, performance isolation, storage-aware caching, and proactive allocation of surplus resources. Zhang *et al.* applied a regression-based approximation of the CPU demand of client transactions on a given hardware [17]. Their experimental results confirm that the regression-based approach provides an efficient capacity planning and resource provisioning of multi-tier applications under changing workload conditions. Nae *et al.* developed a dynamic resource provisioning method for massively multiplayer online games or MMOG operation in data centers [18].

B. Energy Efficiency in Data center

There is a large body of prior studies on optimizing energy efficiency of clusters because of rapidly increasing energy consumption of data centers [19][20]. Total energy used in global data centers in 2010 accounted for approximate 1.5% of total electricity use, for the US that number became 2.2% [21]. Various energy-efficient techniques have been proposed to lower the energy consumption of data centers. For example, Chase *et al.* improved the energy efficiency of server clusters by dynamically resizing active server sets [22]. They provided a solution to respond to power supply disruptions or thermal events by degrading service in accordance with negotiated service level agreements. Rajkumar *et al.* designed energy-efficient resource allocation policies coupled with the scheduling algorithms for managing data center resources for cloud computing [23]. They conducted a set of rigorous performance evaluation; their results reveal that the cloud computing model has immense potential of offering significant cost saving under dynamic workload scenarios. Feller *et al.* evaluated the performance of Hadoop clusters in the traditional model of collocated data and computing services as well as the impact of separating out the services [24]. They conducted an energy efficiency evaluation of Hadoop on physical and virtual clusters in various configurations.

C. Energy-efficient storage clusters

Some important studies have been devoted to the development of energy-efficient storage clusters. For example, Chavan *et al.* proposed a thermal-aware file assignment technique called TIGER aiming to reduce cooling cost of storage clusters. Huang *et al.* developed an energy-saving technique for erasure-coded storage clusters [25]. Ruan *et al.* designed an energy-efficient cluster storage system - ECOS - to improve energy efficiency and perfomrance of storage clusters [26]. Jiang *et al.* built an energy prediction model that estimates the total energy cost of data migrations [27]. In light of the model, they implemented a predictive energy-aware management system, which automatically selects the most energy-efficient method for data transfers.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

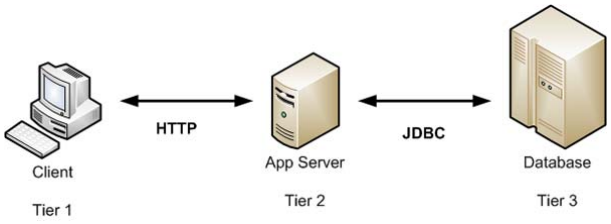


Fig. 1. In the three-tier architecture, the presentation tier allows users to access web pages; the application tier is a logic tier offering services; and the data tier provides data management.

Hardware	
Computer	HP Proliant ML110 G6
CPU	Intel(R) Celeron(R) 450@2.2GHz
Memory	2 GB
Network	1 GigaBit Ethernet network card
Disk	WD-500GB Sata disk([2])
Software	
Operating System	Ubuntu 13.10 Linux Kernel 3.11.0-26
Web Server	Jboss 4.0.2
Database	MySQL 5.1.72

TABLE I. Hardware and software configurations of the testbed

Chai *et al.* designed an energy-efficient method to significantly reduce data migration overhead in storage clusters [28].

D. Thermal efficiency in data centers

A few novel solutions have been proposed to optimize the thermal efficiency of clusters. In 2012, El-Sayed *et al.* conducted a multi-faceted study of temperature management in data centers [29]. A large amount of field data were collected from various production computing environments to investigate the impact of temperatures on the reliability of storage and memory subsystems and server reliability as a whole. Liu *et al.* investigated a prediction scheme to guide a workload management plan for renewable energy as well as computing resources in a data center [30]. Their solution schedules workload and allocates computing resources to improve cooling efficiency [30]. Very recently, Jiang *et al.* conducted thermal profiling of storage servers and developed a thermal model to estimate the outlet temperatures of storage servers [13].In 2010, Pakbaznia developed a power and thermal management engine or PTM, which is employed to determine the number and placement of active servers while simultaneously adjusting supplied cold air temperatures in order to minimize total power consumption for servers and air conditioning units) [31].

Although thermal management and provisioning schemes have been proposed for data centers, little attention has been paid to thermal-aware resource provisioning in data centers. The lack of thermal-aware resource provisioning techniques motivates us to focus on resource provisioning strategies to improve thermal efficiency of data centers where there are changing workload conditions. We aim to develop thermal prediction models to meet the needs of resource provisioning in data centers with a dynamic nature.

III. Experimental Setup

In this section, we first outline our three-tier testbed; then, we discuss the configuration parameters of the testbed as well as the TPC-W benchmark.

The three-tire application architecture shown in Fig. 1 is widely adopted in the industry for the implementation of scalable client-server applications. Such an architecture supports multiple users and can be scaled up under different settings. The presentation tier (a.k.a., tier 1 or client end) is the topmost level of applications, from which users directly access web pages. The application tier (a.k.a., tier 2 or front-end web server) is a logic tier, which is responsible for performing processes of an service. The data tier (a.k.a., tier 3 or back-end database server) is used to house and manage data. We built a three-tier testbed, the hardware and software configuration of which are summarized in Table I.

Request	**Browsing mix**	**Shopping mix**	**Ordering mix**
WIPSb	95.00%	80.00%	50.00%
Home	29.00%	16.00%	9.12%
New Product	11.00%	5.00%	0.46%
Best Seller	11.00%	5.00%	0.46%
Product Detail	21.00%	17.00%	12.35%
Search Request	12.00%	20.00%	14.53%
Search Results	11.00%	17.00%	13.08%
WIPSo	5.00%	20.00%	50.00%
Shopping Cart	2.00%	11.60%	13.53%
Customer Reg.	0.82%	3.00%	12.86%
Buy Request	0.75%	2.60%	12.73%
Buy Confirm	0.69%	1.20%	10.18%
Order Inquiry	0.30%	0.75%	0.25%
Order Display	0.25%	0.66%	0.22%
Admin Request	0.10%	0.10%	0.12%
Admin Confirm	0.09%	0.09%	0.11%

TABLE II. Three workload mixes.

TPC Benchmark W or TPC-W is a web server and database performance benchmark. TPC-W emulates e-commerce transactions where users browse and place orders on a book store web site [32]. In the TPC-W benchmark, there are 14 types of transaction requests (see table II) and 8 data tables. These requests are categorized into two camps, namely, (1) WIPSb - stress tests for web servers and (2) WIPSo - stress tests for database servers. The WIPSb workload is composed of a few database operations and a majority of web page browsing operations. On the other hand, a significant portion of operations in the WIPSo workload are database operations. TPC-W relies on a set of probabilities to simulate the changing behaviors of users. Using the weight of each activity type, TPC-W defines three web workload mixes including web $Browsing$, web $Shopping$, and web $Ordering$.

IV. Resource and Thermal Models

Given access patterns of a web service (i.e., request metadata information), the resource and thermal models are applied to offer an approximation of resource demands and thermal profiles of various transaction types.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

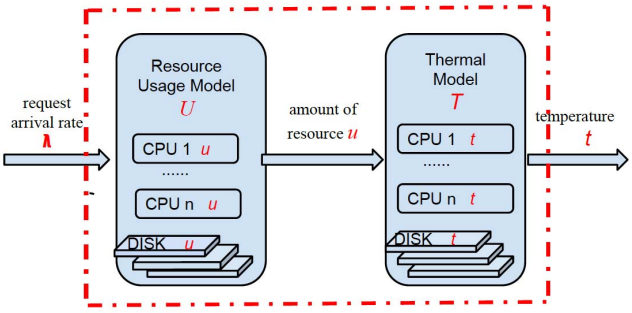


Fig. 2. The resource and thermal models make use of access patterns of a web service to offer an approximation of resource demands and thermal profiles of various transaction types.

Fig. 2 illustrates the correlation between the resource model and the thermal model. More specifically, the resource model relies on access patterns to estimate CPU and disk resource demands for incoming web requests issued to the front-end and back-end servers. Access patterns of any transaction mix are characterized by two factors: request arrival rate λ and the transaction types. Intuitively, resource demands are proportional to arrival rates; a high arrival rate leads to heavily utilized CPU and disk resources. Each transaction type exhibits unique characteristics in terms of resource demands for web and database servers (see also Section V-A). We build a resource usage profile for each transaction type. After individually profiling all the candidate transaction types, we construct a resource model by composing the resource profiles of modified transaction mixes.

With the knowledge of CPU and disk demands offered by the aforementioned resource model, the thermal model yields an approximation of CPU and disk temperatures. Combining the resource model and the thermal model, one can directly derive thermal trends from access patterns. We seamlessly integrate the resource model and the thermal model in a way that we can easily investigate the prediction accuracy of each model. Our preliminary findings show that prediction errors introduced by the resource model can be further magnified the thermal model. In this study, we address this challenge by making TERN adjust model parameters to mitigate prediction errors under dynamically changing workload conditions.

The two models outlined in Fig. 2 have the following three strengths. First, the resource model characterize CPU and disk usages from the arrival rate of each transaction type. Second, the thermal model can speculate the temperature trends of front-end and back-end servers. Third, the models can be integrated to offer thermal profiles of servers in a data center.

V. Modelling

A. Transaction Resource Profiling

We conduct experiments to characterize resource usages as a function of the arrival rate of a transaction type. Resource usages demonstrate resources demands of a transaction mix in front-end and back-end servers.

Table III reveals that all the types of transactions have CPU activities on both the front-end and back-end servers except the *Buy Confirm* transaction type, which exhibits on CPU activity on the back-end server. In addition, the *Buy Confirm* transaction issues no request to disks in the front-end and back-end servers. Among the evaluated transaction types, only the *Customer Reg.* and *Buy Request* transaction types impose disk-intensive workload to the back-end servers. Regardless of arrival rates, no transaction type issues disk requests to the front-end servers.

Request	FE CPU	FE IO	BE CPU	BE IO
Home	✓	✗	✓	✓
New Product	✓	✗	✓	✓
Best Seller	✓	✗	✓	✓
Product Detail	✓	✗	✓	✓
Search Request	✓	✗	✓	✓
Search Results	✓	✗	✓	✓
Shopping Cart	✓	✗	✓	✓
Customer Reg.	✓	✗	✓	✓
Buy Request	✓	✗	✓	✓
Buy Confirm	✓	✗	✗	✗

TABLE III. Resource demands of transaction types.

We observe from Figs. 3(a) and 3(b) that the front-end nodes are more sensitive to the arrival rate than the back-end nodes from the perspective of CPU utilization. For example, the *Buy Confirm* requests impose no CPU activity on the back-end nodes regardless of the arrival rate. When the arrival rate of *Home* requests increases from 10 to 320 No./Sec, the CPU utilization goes up from 5% to 55% on the front-end node, whereas the CPU utilization of the back-end counterparts only changes from 5% to 12%. Fig. 3(c) reveals that the CPU utilization of the front-end and back-end nodes all approximately increase by 8% when the arrival rate reaches 30 No./Sec; in contrast, the disk Utilization climbs up to 95% in this case. We conclude that disks in the back-end nodes are a performance bottleneck that prevents CPUs from being busy on the front-end and back-end nodes even under heavy load (see, for example, Fig. 3(c) when the arrival rate is 33 No./Sec, Figs. 3(a) and fig. 3(b) when the arrival rate is 230 No./Sec and 320 No./Sec, receptively).

Interestingly, Fig. 3(b) shows that the disk utilization of the back-end nodes is initially as higher as 15%, then it drops to 5% when the arrival rate increased to 50 No./Sec. After that point, the disk utilization gradually and slightly decreases even when arrival rate keeps going up to 320 No./Sec. This intriguing trend can be attributed to contents and operations included in *Home* requests. A *Home* request is issued to retrieve a homepage, which is the first webpage of the online book store in the TPC-W benchmark. The homepage contains information of promotional books that triggers database query operations. At an early stage, the disk utilization is high because of retrieving all promotional book information from disks; gradually, the book information loaded from the disks are cached in main memory, thereby making a substantial drop in disk utilization. In order to keep our model simple yet practical, we ignore the impact of this type of transactions on the back-end nodes' disks.

Each type of transactions shows various resource-demanding

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

characteristics on front-end and back-end nodes and; therefore, we individually construct a model for each transaction type. By investigating the relation of resource utilization and arrival rate of the three representative transactions (see Fig. 3), we observe that such a relationship follows linear or quadratic polynomial functions.

B. Resource Usage Profilings

Recall that the linear and quadratic polynomial functions are two candidates to model resource demanding characteristics (see also the previous section). Before incorporate an appropriate function into our models, let us briefly introduce an statistic concept *Residual sum of squares (RSS)* [33].

RSS - the sum of squares of residuals - is a measure of discrepancy between real data and an estimation model. A small RSS value indicates that a model can well fit real data. In (1), U_i^r represents measured CPU or disk utilization when arrival rate is i, U_i^m is CPU or disk utilization estimated at arrival rate i. *RSS*, which describes the sum of discrepancy square, is not a straightforward way of showing each modeled point's deviation from real measured data. To address this problem, we derive the square root of average of RSS ($sraRSS$) $\sqrt{RSS_{average}^{utilization}}$ (see (2)) to quantify the average discrepancy of our model to each measured data at a specific arrival rate. Furthermore, (3) represents a model's accuracy, which is the degree of deviation of average discrepancy between modeled and real data. We refer to $sraRSS$ and R_{sraRSS} as absolute discrepancy and relative discrepancy.

$$RSS = \sum_{i=1}^{n} (U_i^m - U_i^r)^2 \tag{1}$$

$$sraRSS = \sqrt{RSS_{average}^{utilization}} = \sqrt{\frac{\sum_{i=1}^{n}(U_i^m - U_i^r)^2}{n}} \tag{2}$$

$$R_{sraRSS} = \frac{sraRSS}{(\sum_{i=1}^{n} U_i^r)/n} \tag{3}$$

We apply the curve fitting method to obtain the linear and quadratic polynomial function models of all the ten types of transactions. Then, we calculate the absolute discrepancy and relative discrepancy of these models in terms of front-end CPU utilization. The absolute discrepancy values listed in the second column of Table IV show that both the linear and quadratic polynomial functions well fit front-end CPU utilization, the discrepancy of which oscillates only within 3%. The quadratic polynomial models are slightly better than the linear ones; the maximum difference between the two types of models is 0.203% (see *New Product*). For *Home* requests, the difference of the linear model and the quadratic polynomial model is as little as 0.074%, which accounts for about 0.264% to the average of its front-end CPU utilization (see Fig. 3(b)).

We observe from the third column of Table IV that relative discrepancy values of the seven transaction requests are below 15% percentage except for the *Search Result*, *Customer Reg.*, and *Buy* requests. This is because the CPU utilization of these transaction requests in the front-end nodes are relatively low. For example, Fig. 3(c) shows that the front-end node's CPU utilization is under 8%. Comparing the relative discrepancy values of the *Home* and *Buy Confirm* requests in Table IV and front-end CPU utilization in Fig. 3, we discover that high CPU utilization leads to improved model accuracy.

Unfortunately, the above trends are not applicable for disks. Back-end CPU utilization in the case of *Customer Reg.* and *Buy Request* transactions climbs as high as 100%; the relative discrepancy values for both models are larger than 20% (see Table V).

After comparing absolute and relative discrepancies of the linear and quadratic polynomial models for all the ten transaction requests, we decide to apply the linear polynomial functions as our resource usage models, because in this approach estimated CPU and disk utilizations are extremely close to the quadratic functions. In addition, the linear models are less expensive in computing time than the quadratic models. A modern data cent normally has to process thousands of or even millions of requests per second; a computing resource usage model has to predict resource usage for a large number of requests in a timely manner. Low computation overhead of usage models play becomes indispensable for real-time and dynamic resource provisioning.

Our linear models are expressed as (4), where λ_{r_i} is arrival rate of transaction request r_i, $U_{cpu/io}^{r_i}$ is CPU or disk utilization of request r_i when arrival rate is λ_{r_i}. a_{r_i} and b_{r_i} are two coefficients of corresponding transaction request's linear polynomial model. We use the Matlab curve fitting tool to obtain these important coefficients listed in Table VI.

	$sraRSS$		R_{sraRSS}	
	Linear	*Quadratic*	*Linear*	*Quadratic*
Home	2.287	2.213	0.084	0.081
New Product	3.203	3.0	0.121	0.113
Best Seller	3.080	2.958	0.126	0.121
Product Detail	2.121	1.969	0.147	0.136
Search Request	1.967	1.969	0.087	0.087
Search Result	1.896	1.890	0.203	0.202
Shopping Cart	2.372	2.322	0.106	0.103
Customer Register	1.977	1.979	0.363	0.363
Buy Request	1.945	1.944	0.263	0.262
Buy Confirm	1.481	1.406	0.135	0.128

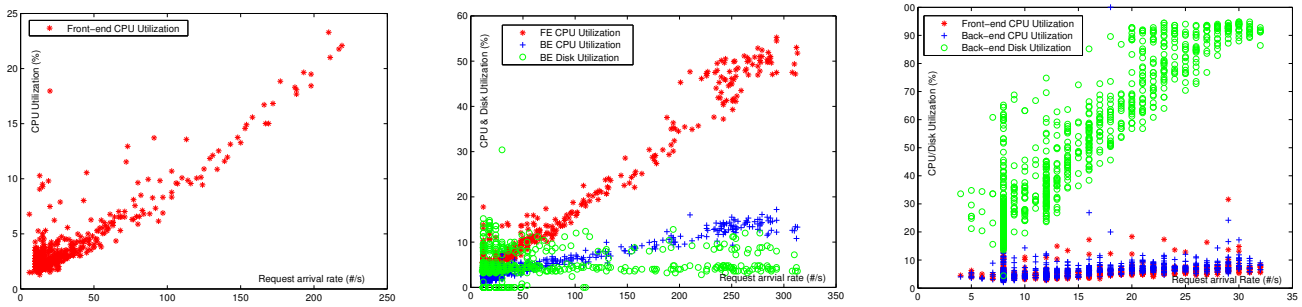
TABLE IV. Absolute and relative discrepancies of the CPU models

	$sraRSS$		R_{sraRSS}	
	Linear	*Quadratic*	*Linear*	*Quadratic*
Customer Register	10.006	9.895	0.208	0.206
Buy Request	12.884	12.728	0.286	0.283

TABLE V. Absolute and relative discrepancies of the disk models

$$U_{cpu/io}^{r_i} = a_{r_i} \times \lambda_{r_i} + b_{r_i} \tag{4}$$

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.



(a) The buy confirm request arrival rate and resource utilization.

(b) The home request arrival rate and resource utilization.

(c) The customer register request arrival rate and resource utilization.

Fig. 3. Request arrival rate and resource utilization in the front-end and back-end nodes

	FE CPU		BE CPU		BE Disk	
	a	*b*	*a*	*b*	*a*	*b*
Home	0.170	1,793	0.044	1.814	—	—
New Product	0.163	3.56	0.057	1.977	—	—
Best Seller	0.119	3.476	0.049	2.036	—	—
Product Detail	0.038	3.935	0.019	2.225	—	—
Search Request	0.096	3.469	0.049	2.212	—	—
Search Result	0.156	3.025	0.098	2.864	—	—
Shopping Cart	0.161	2.234	0.044	1.451	—	—
Customer Register	0.124	3.529	0.184	3.31	3.199	1.609
Buy Request	0.109	2.978	0.056	3.943	0.976	10.78
Buy Confirm	0.084	1.267	—	—	—	—

TABLE VI. Coefficient values of the linear polynomial models.

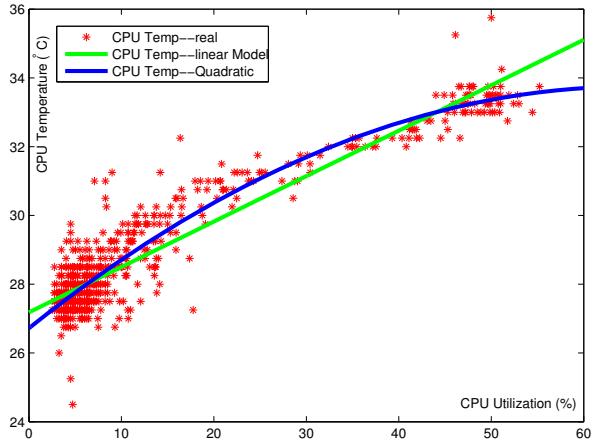


Fig. 4. CPU Thermal Model

C. Thermal Profiling

In this section, we first investigate CPU and disk thermal behaviours in the two groups of experiments. Then, we uncover the relation of CPU/disk temperatures and inlet/outlet temperature gap of a node. In the first group of experiments, we run on the server nodes an multi-thread matrix process, which exhibits CPU-intensive workload. We manipulate CPU utilization of the nodes by tuning request idle time periods.

Fig. 4 plots the average CPU temperature as a function of CPU utilization. The red stars represent real measured CPU temperatures, and the green and blue curves are the linear and quadratic models for CPU temperature estimations. The average CPU temperature is 27 °Cwhen CPU utilizations are 3%. This temperature equals to the CPU internal temperature. Fig. 4 shows that the CPU temperature goes up from 27 °C to 34 °C when the CPU utilization increases from 3% to 55%. The linear model underestimates the CPU temperature with CPU utilization ranging from 15% to 45%. In contrast, the quadratic model has better fitting performance than the linear model during this CPU utilization range.

The absolute discrepancy of the liner and quadratic models are 0.66 and 0.61 °C, respectively. The discrepancy difference of the two models is 0.05 °C, meaning that the quadratic model is 7.5% more accurate than the linear model. Nevertheless, for all the ten types of transactions, the quadratic model is only on average 3% more accurate than the linear model. As a consequence, while making a tradeoff between accuracy and computing overhead, we choose the linear model for estimating transaction request resource usage and the quadratic model for predicting CPU temperatures.

The CPU thermal model is expressed as (5), where T_{r_i} is CPU temperature when CPU Utilization is $U^{r_i}_{cpu/io}$, and *a, b, c* are three coefficients.

$$T_{r_i} = a \times {U^{r_i}_{cpu/io}}^2 + b \times U^{r_i}_{cpu/io} + c \tag{5}$$

where a=-0.001541, b=0.2076, c=27.86.

Now we conduct a thermal profiling test on disks in the computing nodes. In this group of experiments, we create disk-intensive workload using IOzone - a filesystem benchmark. By gradually stressing disks, we measure disk temperatures when disk utilization increases. Fig. 5 shows the impact of disk utilization on the average disk temperature in the cluster of nodes. We observe that the disks are less thermal sensitive than the CPUs in the nodes. Besides, the relationship between disk temperatures and utilization does not match neither the

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

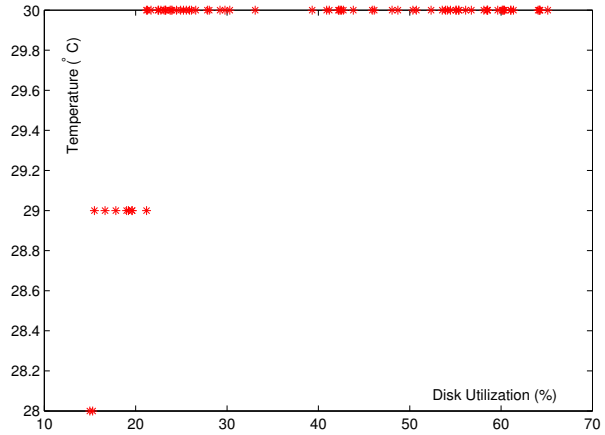


Fig. 5. Disk Thermal Profiling.

linear nor quadratic models. However, the piecewise function (see Equation (6)) can be used to depict the relation.

$$T_{r_i} = \begin{cases} 28\ ^{\circ}\text{C} & U_{io} \in (0, 15) \\ 29\ ^{\circ}\text{C} & U_{io} \in [15, 25] \\ 30\ ^{\circ}\text{C} & U_{io} \in (25, 100) \end{cases} \tag{6}$$

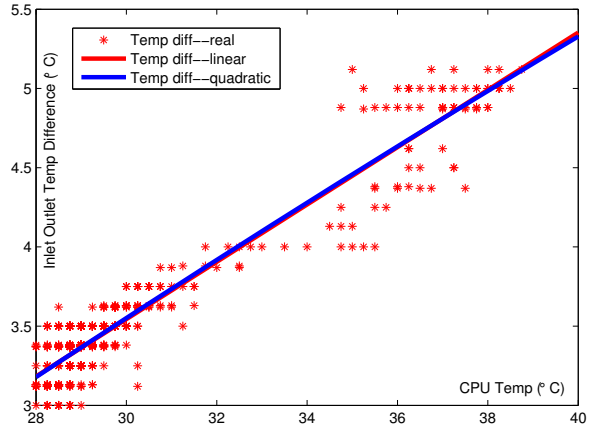


Fig. 6. Inlet and Outlet Temperatures

$$T_{r_i}^{inlet/outlet} = a \times T_{cpu} + b \tag{7}$$

where a=0.1809, b=-1.883

D. Integrated Models

In the previous sections, we have modeled relations between each transaction type's arrival rate and CPU and disk utilizations (see (4)) for CPU utilization, (see (5)) for thermal modeling. (7) shows the relation of inlet/outlet temperature difference and CPU temperature.

Integrating (6), (5), and (4), we build a thermal model in form of a function of the requests arrival rate as

$$T_{r_i} = f(\lambda_{r_i}) \tag{8}$$

When putting (4), (5),and (7) together, we obtain an inlet/outlet temperature difference model as a function of arrival rate below.

$$T_{r_i}^{inlet/outlet} = f(\lambda_{r_i}) \tag{9}$$

In Section III, we describe three types of workload mixes, each of which has a specific percentage combination of individual transaction requests. Therefore, CPU and disk utilization model of a workload mix can be derived as

$$U_{mix} = \sum_{r_i \in R} P_{r_i} \times U_{r_i} \tag{10}$$

where P_{r_i} is transaction request r_i's combination percentage in the workload mix and U_{r_i} is the transaction request's resource usage model.

Similarly, we construct the thermal model of a workload mix as (11) by incorporating the CPU and disk thermal models of multiple transaction types. (12) will be talked in Section VII-B.

$$T_{mix} = \sum_{r_i \in R} P_{r_i} \times T_{r_i} \tag{11}$$

$$T_{r_i}^{adjusted} = T_{r_i} \pm \sigma^{AD} \tag{12}$$

VI. A Self-Adjustment Strategy

A. Deriving Algorithm

The CPU temperature model and the inlet/outlet temperature difference model proposed in the previous section do not fit well to real data when CPU temperature is in the range between 33 °C to 37 °C. In this section, we develop a self-adjustment strategy (see Algorithm 1) to automatically adjust the models according to the models' absolute and relative discrepancies.

In the self-adjustment strategy, we partition a model's CPU/disk temperature range into an array of small consecutive ranges. We set initial absolute and relative discrepancies to 0, and set the initial model to using the original model (see Lines 1-10). Next, in an iterative process, if previous CPU/disk temperature range has a relative discrepancy exceeding the specified threshold, then the model will self-adjust by increasing or decreasing previous range's absolute discrepancy depending on if the model's overestimation or underestimation. If the model is accurate for the current range, there will be no need to adjust the model's parameters. Obviously, the model applied for the first range is the original model where $RD_0 = 0 < RD_{threshold}$ (see Lines 12-16). In the last step, the algorithm calculates the current range's absolute and relative discrepancies and determines whether or not the model should adjust itself for the next range. Then, the algorithm repeatedly process the next range until all the ranges are processed. (see Lines 17-21). The model powered by the self-adjustment strategy is comprised of multiple models for all the individual ranges.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

Input: $\langle M, X_{min}, X_{max}, R, AD, RD^{threshold}, N\rangle$
M: original model
X_{min}: minimum X value of Model M
X_{max}: maximum X value of Model M
R: real measured data
AD: Model M's overall absolute discrepancy
$RD^{threshold}$: predefined relative discrepancy
N: the amount of divided ranges of x axis
Output: $M^{adjusted}$ the adjusted model

if $N = 1$ **then**
  $sign = \sum_{X_{min}}^{X_{min}+X_{max}}(R–M)$
  $M^{adjusted} = M + sign \times AD$
else
  $X_{incre} \leftarrow (X_{max}–X_{min})/N$
  $sign = 1$
  $i = 0$
  $AD_i = 0$
  $RD_i = 0$
  $M_i = M$
  while $X_{min} < X_{max}$ **do**
    if $RD_i > RD_{threshold}$ **then**
      $M_{i+1} = M_i + sign \times AD_i$
    else
      $M_{i+1} = M_i$
    end
    $i \leftarrow i+1$
    $RD_i = R_{sraRSS}^{X_{min}+X_{incre}}{}_{X_{min}}$
    $sign = \sum_{X_{min}}^{X_{min}+X_{incre}}(R–M)$
    $AD_i = sraRSS_{X_{min}}^{X_{min}+X_{incre}}$
    $X_{min} = X_{min}+X_{incre}$
  end
end
return $M^{adjusted} = \langle M_1, M_2, \ldots, M_N\rangle$

Algorithm 1: A SELF-ADJUSTMENT STRATEGY

B. Evaluate of Self Adjustment

In this section, we apply Algorithm 1 to model (7), where N is set to 11 and the range of CPU temperature is set to 1. For instance, when CPU temperature is 28 °C, we apply the original model to estimate thermal behavior as well as absolute and relative discrepancies. Then, these parameters can be adjusted for the next CPU temperature range from 29 to 30 °C.

Fig. 7 shows the absolute discrepancies of the three models (i.e., the original mode, the model with self-adjustment where the predefined relative discrepancy threshold is set to 5% and 3%, respectively). Fig. 8 shows the corresponding relative discrepancies. Absolute discrepancies plotted in Fig. 7 are below 0.15 from 28 °C to 35 °C for all the three models. Then, the discrepancies jump from 0.125 to 0.3 when CPU temperature increase from range (33,34) to range (34,35). This trend is consistent with that shown in Fig. 6.

For the self-adjustment strategy where the threshold is set to 5%, the relative discrepancy increases from 0.032 to 0.07 when the range moves from (33,34) to (34,35). In this case, the model starts adjusting the parameters. As a result, the absolute discrepancy of the mode in in range (35,36) decreases from 0.38 to 0.35 (see Fig. 8) and the relative discrepancy is reduced from 0.085 to 0.078, which is approximately 10% more accurate than the original mode. From range (36, 37) to (37, 38), the 5%-threshold model also reduces absolute discrepancy. However, the 3%-threshold model begins to adjust itself at range (29,30) in which the absolute discrepancy noticeably increases. Additionally, its absolute discrepancy are larger than those of the original model and the 5%-threshold model in ranges (36,37) and (37,38).

The models with our self-adjustment strategy improve estimate accuracy of the traditional modeling approach. The new models' accuracy is affected by the predefined relative discrepancy threshold. Specifically, if this threshold is set too high, the model will degraded to its original model. On the other hand, if the threshold is set too low, the self-adjustment strategy will be very sensitive to relative discrepancies. The self-adjustment strategy coupled with a low threshold is inadequate for models in which relative discrepancies are frequently and dramatically changing.

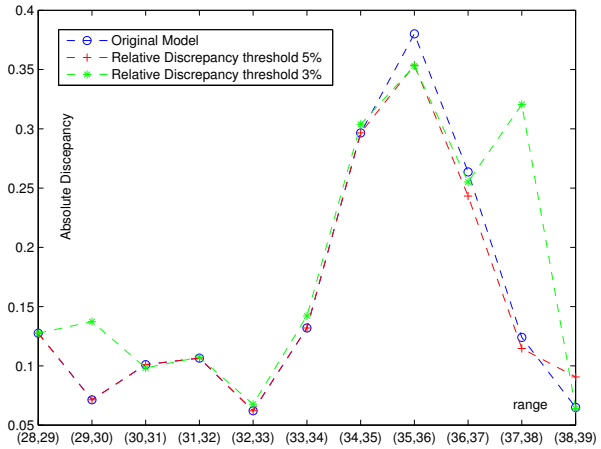


Fig. 7. Absolute discrepancy

VII. MODEL VALIDATION

In this section, we validate the accuracy of the resource usage models as well as the thermal models in TERN. Section VII-A evaluates the resource usage models using three representative transaction requests (i.e., *Home*, *New Products* and *Buy Request*). Then, we apply the resource usage models to predict resource utilization in real-world scenarios of three workload mixes. We discuss in Section VII-B the validation of the thermal models for various transaction requests in addition to the three workload mixes.

A. Resource Usage Model validation

Fig. 9 shows that regardless of the three types of transactions, our resource usage models accurately estimate resource demands of front-end CPUs, back-end CPUs, and back-end disks.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

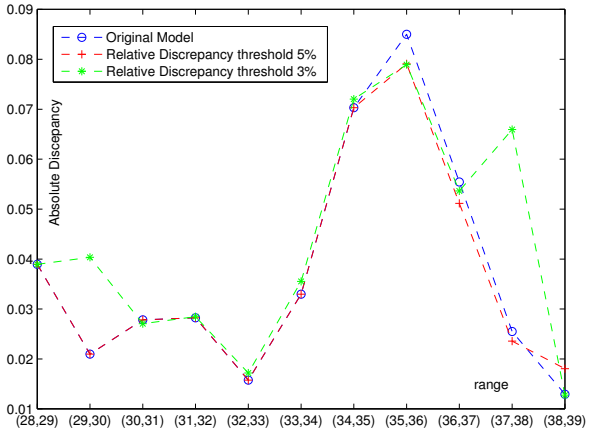


Fig. 8. Relative discrepancy

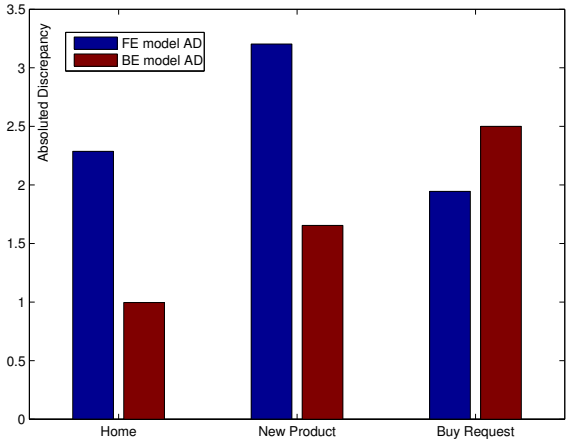


Fig. 10. Absolute discrepancies of the resource usage models in the three types of transactions.

For the *Home* request, the model slightly overestimates front-end CPU usage by less than 3% when arrival rate increases from 10 No./Sec. to 130 No./Sec. And, the overestimation goes up to approximately 5% when arrival rate is in the range between 130 No./Sec. to 250 No./Sec. On the other hand, the model underestimates front-end CPU usage by anywhere between 5% to 10% when arrival rate increases from 250 No./Sec. to 300 No./Sec.

We observe from Fig. 9 that in the case of *Home* request, the back-end CPU model is superior to the front-end CPU model. A similar trend can be found for the resource usage models in the cases of *New Products* and *Buy Request* requests. In the *Buy Request* case, the disk resource usage model has a high absolute discrepancy of 13. Nevertheless, Fig. 9(c) reveals that real measured data are evenly distributed among the estimated data except for the cases where arrival rate is below 30 No./Sec.

Fig. 10 illustrates the absolute discrepancies of the models in the three transaction types. For the *Home* transaction request, the absolute discrepancy of the back-end nodes is more than 2 fold of that of the front-end node. A similar observation can be drawn for the *New Product* transaction request. Unlike the models for the *Home* and *New Product* requests, the *Buy Request* model's absolute discrepancy of back-end nodes is larger than than that of front-end one; this pattern is not intuitively observed in Fig. 9(c).

B. Performance of the Thermal Models

Recall that the thermal models proposed in Section V-D (see also (8)) make use of arrival rate to predict CPU temperatures. In this subsection, we focus on evaluating the prediction performance of the thermal models. Figs. 12 shows comparisons between predicted CPU temperatures and the measured ones when the arrival rate is gradually increased.

The experimental results suggest that the thermal models can predict CPU temperatures of the nodes processing the *Shopping Cart* transaction requests under a wide range of arrival rate (see fig 13(a)). Figs.13(b) and 13(c) show that for the two transaction requests, the models either overestimate or underestimate CPU temperatures of the front-end and back-end nodes. For example, in the $Home$ request case, the models overestimate the temperatures by 1.23 °C and 1.46°C for the front-end and back-end nodes, respectively. On the other hand, the models underestimate the CPU temperatures of the front-end and back-end nodes by 0.71 and 0.70 °C for the *Product Detail* workload.

We adjust the models (see (12) in Section V-D) to optimize the prediction accuracy of the thermal models, where parameter N is set to 1 and σ^{AD} equals to a corresponding absolute discrepancy (i.e., an underestimation or overestimation value). Fig. 13 illustrates that adjusted models are more tight fitting to real measured temperature.

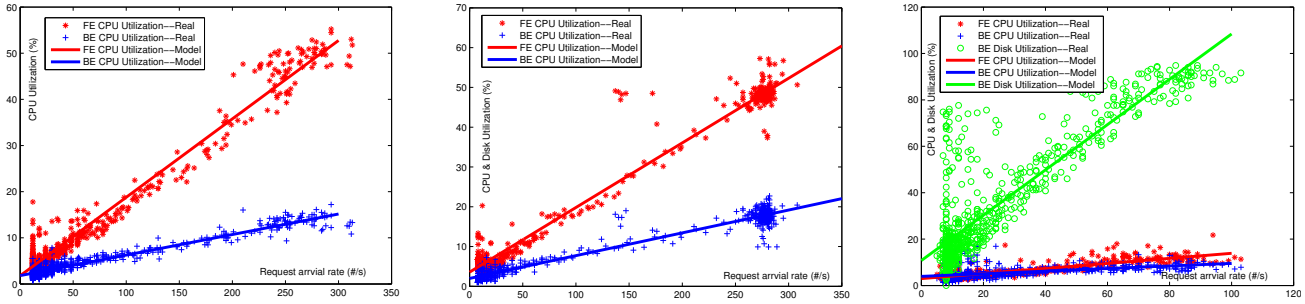
C. Thermal model evaluation for workload mix

Figure 14 shows the performance of three types of workload mix's thermal models (see (11)) on front-end and back-end. Workload *Browsing*'s thermal model (see Fig. 14(a)) slightly underestimate CPU temperature in front-end when arrival rate increase from 10 No./Sec to 170 No./Sec, then it fits well when arrival requests climbs to 350 No./Sec, Similar trend exists on its back-end. *Shopping* and *Ordering* workload's thermal model perform a little better than *Browsing* model does.

VIII. CONCLUSION

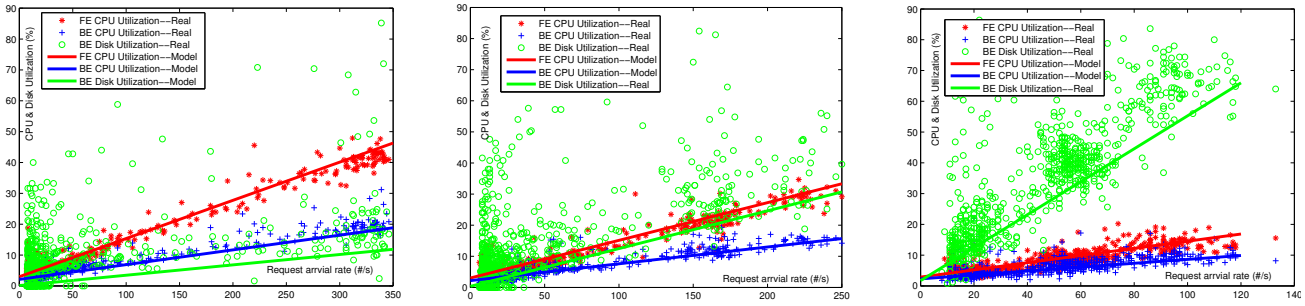
We have proposed a modeling system called TERN, which dynamically adjust itself to predict thermal behaviors of hardware resources allocated to applications running on clusters. TERN is conducive to predicting thermal patterns of diverse workload conditions with changing transaction mixes. More precisely, TERN seamlessly integrates a resource utilization model and a thermal model, which establish correlation between resource utilization and thermal trends of processors and hard drives in servers. We made use of the TPC-W benchmark to conduct thermal profiling studies of workload with various

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.



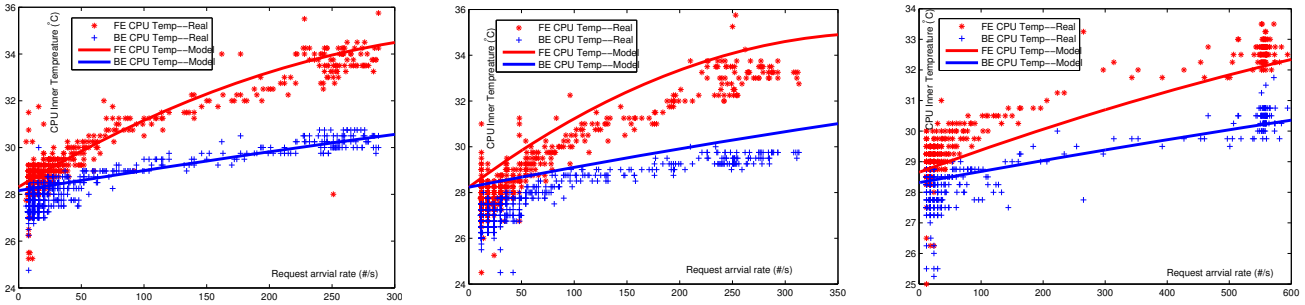
(a) *Home* (b) *New Product* (c) *Buy Request*

Fig. 9. Accuracy of the resource usage models for front-end CPUs, back-end CPUs, and back-end disks in the cases of the *Home*, *New Product*, and *Buy Request* transactions.



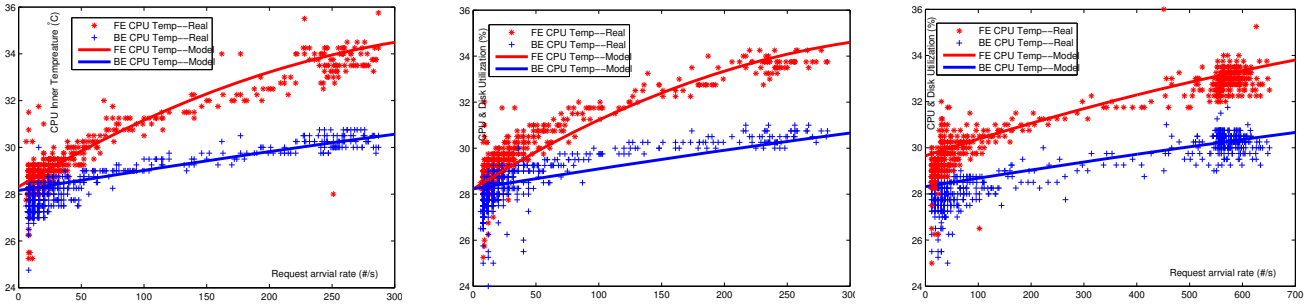
(a) *Browsing* (b) *Shopping* (c) *Ordering*

Fig. 11. Workload mix's CPU & Disk Utilization



(a) *Shopping Cart* (b) *Home* (c) *Product Detail*

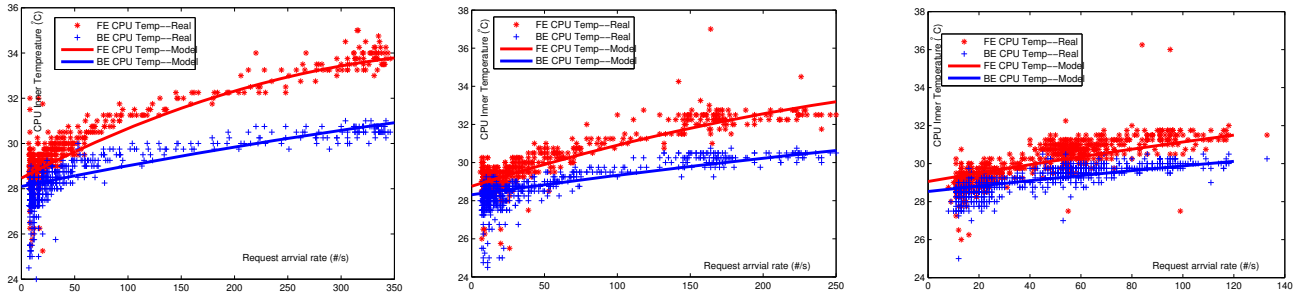
Fig. 12. Individual transaction request's CPU inner temperature



(a) *Shopping Cart* (b) *Home* (c) *Product Detail*

Fig. 13. Individual transaction request's CPU inner temperature

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.



(a) *Browsing* (b) *Shopping* (c) *Ordering*

Fig. 14. Workload mix's CPU inner temperature

transaction mixes. A salient feature of TERN lies in its capability of adjusting the modeling parameters to improve prediction accuracy by approximately 10% for changing load patterns. We showed through the experiments performed on a real-world cluster that TERN offers a simple yet powerful solution for thermal-aware resource provisioning in data centers.

The main drawback of TERN is that it is focused on CPU and disk resources. For large-scale data centers, we have to consider other shared system resources like main memory and network interconnections. We plan to extend TERN by incorporating the other resource types supporting various applications on clusters. Furthermore, we will exploit a prediction model for thermal-aware Hadoop clusters in the context of big data analytics.

ACKNOWLEDGMENTS

Xiao Qin's work is supported by the U.S. National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0917137 (CSR), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0831502 (CyberTrust), CNS-0855251 (CRI), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), and DUE-0830831 (SFS).

REFERENCES

[1] B. G. H. T. S. A., "Beyond the data deluge," in *Computer Science*, vol. 323, March 2009, pp. 1297–1298.

[2] J. T., "Slow but steady growth for data centers through 2015," October 2011.

[3] C. L. B. P.E., "In the data center, power and cooling costs more than the it equipment it supports," in *Electronics Cooling*, February 2007.

[4] C. Bash and G. Forman, "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center," in *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, ser. ATC'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 29:1–29:6.

[5] J. Moore and J. S. Chase, "Weatherman: Automated, online, and predictive thermal mapping and management for data centers," in *In International Conference on Autonomic Computing*, 2006.

[6] A. Manzanares, X. Qin, X. Ruan, and S. Yin, "Pre-bud: Prefetching for energy-efficient parallel i/o systems with buffer disks," *ACM Transactions on Storage (TOS)*, vol. 7, no. 1, p. 3, 2011.

[7] M. Allalouf, Y. Arbitman, M. Factor, R. I. Kat, K. Meth, and D. Naor, "Storage modeling for power estimation," in *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, ser. SYSTOR '09. New York, NY, USA: ACM, 2009, pp. 3:1–3:10.

[8] Y. Pan, R. Yin, and Z. Huang, "Energy modeling of two office buildings with data center for green building design," *Energy and Buildings*, vol. 40, no. 7, pp. 1145 – 1152, 2008.

[9] Z. Zong, A. Manzanares, X. Ruan, and X. Qin, "Ead and pebd: two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters," *Computers, IEEE Transactions on*, vol. 60, no. 3, pp. 360–374, 2011.

[10] R. E. Brown, E. R. Masanet, B. Nordman, W. F. Tschudi, A. Shehabi, J. Stanley, J. G. Koomey, D. A. Sartor, and P. T. Chan, "Report to congress on server and data center energy efficiency: Public law 109-431," 06/2008 2008.

[11] J. G. Koomey, "Estimating total power consumption by servers in the u.s. ad the world," Lawrence Berkeley National Laboratory and Consulting Professor, Stanford University, Tech. Rep., 2007.

[12] C. D. Patel, "A vision of energy aware computing from chips to data centers," in *The International Symposium on Micro-Mechanical Engineering*, 2003.

[13] X. Jiang, M. I. Alghamdi, J. Zhang, M. Assaf, X. Ruan, T. Muzaffar, and X. Qin, "Thermal modeling and analysis of storage systems," in *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*. IEEE, 2012, pp. 31–40.

[14] B. Urgaonkar, P. Shenoy, A. Chandra, P. Goyal, and T. Wood, "Agile dynamic provisioning of multi-tier internet applications," *ACM Trans. Auton. Adapt. Syst.*, vol. 3, no. 1, pp. 1:1–1:39, Mar. 2008.

[15] "Amazon 2000," http://www.fool.com/news/2000/wmt001127.htm.

[16] R. P. Doyle, "Model-based adaptive resource provisioning in a web service utility," Ph.D. dissertation, Durham, NC, USA, 2003, aAI3135127.

[17] Q. Zhang, L. Cherkasova, and E. Smirni, "A regression-based analytic model for dynamic resource provisioning of multi-tier applications," in *Autonomic Computing, 2007. ICAC '07. Fourth International Conference on*, June 2007, pp. 27–27.

[18] V. Nae, A. Iosup, and R. Prodan, "Dynamic resource provisioning in massively multiplayer online games," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 380–395, March 2011.

[19] L. A. Barroso and U. Hlzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, 2007.

[20] D. J. Brown and C. Reams, "Toward energy-efficient computing," *Commun. ACM*, vol. 53, no. 3, pp. 50–58, Mar. 2010.

[21] J. Koomey, *Growth in data center electricity use 2005 to 2010*. Oakland, CA: Analytics Press, 2011.

[22] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," in *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles*, ser. SOSP '01. New York, NY, USA: ACM, 2001, pp. 103–116.

[23] R. Buyya, A. Beloglazov, and J. H. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," *CoRR*, vol. abs/1006.0308, 2010.

[24] E. Feller, L. Ramakrishnan, and C. Morin, "Performance and energy efficiency of big data applications in cloud environments: A hadoop case study," *Journal of Parallel and Distributed Computing*, 2015.

[25] J. Huang, F. Zhang, X. Qin, and C. Xie, "Exploiting redundancies and deferred writes to conserve energy in erasure-coded storage clusters," *ACM Transactions on Storage (TOS)*, vol. 9, no. 2, p. 4, 2013.

[26] X. Ruan, S. Yin, A. Manzanares, J. Xie, Z. Ding, J. Majors, and X. Qin, "Ecos: An energy-efficient cluster storage system," in *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*, Dec 2009, pp. 79–86.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.

[27] X. Jiang, J. Zhang, M. Alghamdi, X. Qin, M. Jiang, and J. Zhang, "Peam: Predictive energy-aware management for storage systems," in *Networking, Architecture and Storage (NAS), 2013 IEEE Eighth International Conference on*, July 2013, pp. 105–114.

[28] Y. Chai, Z. Du, D. A. Bader, and X. Qin, "Efficient data migration to conserve energy in streaming media storage systems," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 11, pp. 2081–2093, 2012.

[29] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder, "Temperature management in data centers: why some (might) like it hot," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, pp. 163–174, 2012.

[30] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1. ACM, 2012, pp. 175–186.

[31] E. Pakbaznia, M. Ghasemazar, and M. Pedram, "Temperature-aware dynamic resource provisioning in a power-optimized datacenter," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '10. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2010, pp. 124–129.

[32] "Tpc-w benckmark specification," http://www.tpc.org/tpcw/spec/tpcw_v1.8.pdf.

[33] H. S. Norman R. Draper, *Applied Regression Analysis*, 3rd ed. Wiley-Interscience, April 1998.

Authorized licensed use limited to: Auburn University. Downloaded on November 17,2026 at 23:01:28 UTC from IEEE Xplore. Restrictions apply.