This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

Optimal Performance-Aware Cooling on Enterprise Servers

Christine S Chan, Alper Sinan Akyürek, Baris Aksanli, *Member, IEEE* and Tajana Šimunić Rosing, *Fellow, IEEE*

***Abstract*—Datacenters house massive databases and applications to provide business decision support and cloud services where commercial success is contingent on timely responses. The servers that these tasks run on dissipate a lot of power, requiring equally powerful cooling systems to maintain a safe and efficient temperature level. In a typical enterprise server, server chassis fans can generate vibrations that are powerful enough to degrade the performance of data-intensive workloads. Our methodology measures and reproduces real-life vibrations on a rack server to evaluate the performance of different hard disks. Effective hardware management relies on an accurate understanding of these devices and their interactions to mitigate any performance degradation and meet thermal constraints. While current strategies focus on managing processing resources, at the expense of more data-dependent workloads, our work approaches server efficiency by targeting the cooling-performance relationship in conjunction with other dependencies between power, thermal, and cooling. We extract a model from common database benchmarks based on expected resource utilization and corresponding cooling needs, while considering these mechanical disturbances. Our proposed strategy uses convex optimization to maintain thermal constraints at all times, while reducing the energy consumption of a server by 65% compared to basic PID controllers, or by 19% in comparison to advanced hardware management techniques proposed in literature.**

***Index Terms*—server modeling, database performance, vibrations, thermal management, fan cooling, convex optimization.**

I. Introduction

ENTERPRISE servers generate profit for their operators by delivering data and computing for a wide range of concurrent applications at high performance. They serve simultaneous requests from multiple clients (e.g. in big data and cloud computing services) while guaranteeing a quality of service (QoS) for each one. The QoS metric for different applications may vary - e.g. data throughput for online transaction processing (OLTP) database operations [1], or response time for interactive web applications, but as developments in processing power have advanced far ahead of storage and communication, many high-performance services are now I/O bound. Their datasets are primarily stored on traditional disk media, and partially cached in solid state drives and physical memory [2]. Datacenters rely on many drive types to maintain these different tiers of storage. - e.g. the most mission-critical data is stored in Tier 1, the majority of current business data is in Tier 2, and archival storage is in Tier 3 [2]. Tier 2 currently comprises the largest proportion of current data that needs to be readily accessible (i.e. non-archival), and is commonly fulfilled by either Serial Attached SCSI (SAS) or Serial ATA (SATA) drives. They differ greatly in manufacturing tradeoffs in terms of mechanics, materials and electronics [3] - the more robust SAS drives are reserved for more expensive deployments. There are emerging storage strategies that bypass hard drive performance issues to recover high access speeds, such as RAMCloud [4], which divides and distributes datasets across physical memory in multiple machines. However, even these approaches still ultimately rely on hard drives for large scale data storage.

To maintain the integrity of hardware components, processors manage workload scheduling and on-chip thermal management dynamically, while powerful server chassis fans work in combination with the building HVAC or passive heat removers to maintain a thermal set point [5]. The power consumption of these fans grows cubicly with the speed settings [6]. There are efforts to reduce datacenter cooling power by reconfiguring rack organization (e.g. hot/cold aisles [7]), creative chillers, and task allocation across multicore processors [8] and even room placement [9]. In an individual server, the largest power consumers are the processor chip and the cooling subsystem (we measured 37% and 29% respectively). In a typical datacenter of 20,000 servers at a 1.5 power usage effectiveness (PUE), 24% of its monthly budget goes towards the utility bill [10]. Any improvement in the server and auxiliary energy consumption (cooling, etc) can dramatically lower power budgets, improve service reliability in case of power instabilities, and ultimately improve profit margins for the datacenter operator [11].

We focus on a mainstay of server workloads - database services. Database software is written assuming that underlying hardware resources including CPU cycles, memory access and IO bandwidth are fully available. However, the operating context may limit this resource availability - e.g. power caps or thermal constraints are aspects of the physical environment that can limit software application behavior. In this work, we address a critical source of data performance degradation that is often neglected. Mechanical disturbances generated by the cooling system can cause temporary crashes or misses in spinning storage systems, which in turn inflate workload execution times and server uptime electric bills [12]. Even small disk latencies can cascade into large effects on final application performance - 5% disk latency can lead to over 40% slowdown in the total performance [13], while others have measured a 60% disk delay leading to 170% final delay

Christine Chan is with Qualcomm Technologies Inc, San Diego, CA 92121 USA e-mail: csc019@eng.ucsd.edu

Baris Aksanli is with the Electrical and Computer Engineering Department, San Diego State University, CA 92182 USA e-mail: baksanli@sdsu.edu

Alper Sinan Akyurek and Tajana Simunic Rosing are with the Department of Computer Science and Engineering, University of California San Diego, CA, 92093 USA e-mail: aakyurek,tajana@eng.ucsd.edu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

in a database query execution [12]. These transient problems can be very difficult to diagnose in deployment or to replicate in a lab setting without the correct surrounding environmental factors. They are also neglected by existing thermal models and management policies.

Since current enclosure, cabinet and raised-floor room designs cannot fully eliminate vibrations [13], we turn to software-based detection and control. Enterprise servers already have a side-band "service processor" to monitor hardware sensors, execute power management, and log maintenance events, accessed via the Intelligent Platform Management Interface (IPMI). This would be an appropriate platform to detect vibrations from the fan controller and respond accordingly. Orthogonal to mechanical upgrades, a software update allows for fast, low-cost adaptation.

Fortunately, though database workloads can be highly demanding of the hardware platform and increasingly complex in optimization, they are also fairly well-known and predictable at scale [14]. By modeling the workloads in terms of their resource consumption, server operators can predict the application's needs and manipulate the operating conditions such as temperature and core availability to improve performance. Most current strategies focus on manipulating processing resources such as multi-core task scheduling and frequency scaling. Notably, our work approaches the server efficiency problem by targeting the cooling-performance relationship. Our results are based on real physical telemetry of a late-model multi-threaded, multi-core server processor running a standard database benchmark suite (TPC-H [15]). The proposed server model and control policy demonstrate up to a 3x speed up over state of the art policies, leading to 19-65% energy reduction while still meeting thermal constraints.

II. Related Work

Here, we review three main research areas that around server data performance as supported by hard disk drives. First, we identify database applications representations in terms of hardware utilization and data accesses. Second, we discuss existing physical and mechanical designs that affect hard drive performance in datacenters. We close by summarizing state of the art power, thermal and cooling policies for servers, and discuss our contributions to the area.

A. Database Workload Modeling

Database performance can be quantified and analyzed many different ways. End-to-end metrics such as total execution time are used to signal critical failures or crisis status [14]. For a more detailed understanding, applications can be divided into phases of software demands and elemental operations, but the complexity of database platforms necessitate the use of machine learning techniques rather than relying on expert design [16]. To predict total execution time of separate database queries using design-time characteristics, some have found that clustering techniques out-perform regression for multi-variate feature sets [17]. An orthogonal method of representing workloads is to inspect their interactions with hardware resources, which lends more naturally to hardware management policies. For example, a particular query behavior can be described with microarchitectural statistics (e.g. IPC and cache-miss), and transitions between behavior can be modeled with a Markov decision process, leading to thermal management decisions [18]. A query can also be described in terms of the size, location, and frequency of disk accesses [19], [20]. These policy solvers choose optimal execution plans assuming ideal drive performance; if drive throughput is affected by external vibrations, solver results may no longer be naccurate.

B. Mechanical considerations: shock and vibration

Vibrations and shock can have significant detrimental effects on hard disk drive operation [21], [22]. Many server vendors and large customers have made design improvements in drive enclosures [3], [23], server chassis [24], racks [25], and even raised-floor rooms [13], [26], to mitigate vibrations and preserve drive data integrity. These improvements include sturdier material choices, physical re-organization of vibration sources (fan arrays, hard drives), and signal processing to cancel sensed vibrations. The vibration protections only target well-known sources such as the spinning hard drive motors themselves, physical drops, and HVAC building cooling systems [27]. Their success metrics are geared towards lowering hard drive failure rates, generally caused by head crashes (i.e. when the read-write head makes contact with the disk platters, causing irreverible damage) [28]. Liquid cooling [29], [30] would reduce mechanical disturbances to the system, but are prohibitively expensive for today's commodity systems. To our knowledge, there are no solutions currently in the market that account for the persistent, dynamically changing vibrations generated from fan cooling within the server. Concurrently, there are no metrics that quantify vibrational effects in terms of instantaneous but non-lasting drive performance degradation.

C. Power, Thermal and Cooling Management

The largest power consumers in servers are the processor chip and cooling subsystem. Fans have a cubicly growing motor power consumption profile [6], while processor leakage power grows quadratically with increasing temperature (i.e. lower fans). For a fixed workload, there is a single optimal point where some fan speed achieves the lowest combined processor leakage power and fan motor power [31]. High, fluctuating temperatures are correlated with poor drive reliability [32], [33]. We show that the observed disk performance degradation is likely due to interactions with the cooling system, and not temperatures *per se*. To reduce the thermal load, the processor can gate the clock or perform dynamic voltage and frequency scaling (DVFS), at the cost of direct reduction in performance [34]. In some workloads, frequency scaling may have unexpected effects - what is optimal from the core's perspective may not yield desirable results for the larger system if the performance bottleneck is actually elsewhere [35]. For cooling, a standard industrial policy proportional-integral-derivative (PID) control [36], which some newer solutions are based on [37]. Task assignment can be done with some awareness of datacenter physical layout [38], [39], but these techniques fail to account for the cooling interactions at the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

lower level of server fans. Traditionally, the thermal effect on performance is only measured in terms of core compute speed [34], even by studies of disk-heavy database query performance [40]. We assert that even when thermal is not considered an issue by conventional standards (e.g. high temperature), data performance can still suffer, because cooling has a significant side-effect - existing work has shown that internal server fans can negatively impact drive throughput by anywhere between 60-88% [12], [41]. A comprehensive understanding of the system enables model-predictive control to maintain a stable system state and make guarantees about system behavior. Relevant models include thermal circuit simulators [42], time-based temperature predictors [43] [44] or workload-based temperature prediction [37]. Well-defined hardware configurations and operating ranges lend themselves to efficient and stable control-theoretic solutions for cooling decisions [45], [46]. While these strategies leverage the trade-offs between cooling power consumption and core speed, they neglect the relationship between cooling and data performance. Thus, they may yield subpar performance for data-intensive workloads.

In comparison to solutions in the current state of the art, our contributions are three-fold:

- We *quantify fan-disk interactions* that cause difficult-to-diagnose performance degradation in data-intensive workloads. Our measurements of a real operating datacenter and lab reproductions show up to a 88% hit on disk write throughput when fans are at their maximum setting.
- We develop a *model of a server to represent dependencies between server performance and physical effects*, including power, thermal and cooling. Using analytical models as opposed to conventional simulators, we enable formal optimization of the overall system.
- We use convex optimization to design *a proactive policy for optimally efficient fan management*. Compared to existing controllers and those proposed in literature, our model-predictive control yields provably higher energy savings (up to 80%) and faster workload completion times (up to 70%) while meeting temperature constraints.

The rest of this document is organized as follows: Section III documents our measurements of cooling and performance interactions in a real, operating datacenter server. In Section IV, we develop a system model based on physical measurements of thermal, cooling and disk performance. Section V formulates and solves the hardware thermal management problem optimally for runtime energy. Finally in Section VI, we evaluate how the optimal hardware management policy performs as compared to current state of the art.

III. Cooling and performance interactions

We present a methodology for characterizing any server disk's response to vibrations that it may encounter in a typical datacenter. Our parametric characterization suite of experiments measures the vibrational sensitivity of a diverse set of disks. In all experiments, the ambient temperature is tightly controlled, isolating any drive performance effects to mechanical sources.

TABLE I
Test server specifications

Processor	8 cores @ 3.0GHz, 40nm
Memory	16 x 16GB DIMM
Operating system	Solaris 11.1, firmware 8.2.1
DBMS	Oracle 11.2.0.3
Idle processor power	75W
Idle server power	267W
Typical server power range	330-600W
Maximum air flow	145 cubic feet per minute (cfm)
Maximum fan power	180W
Room temperature	25°C
Chassis internal temperature	30°C

A. Measurement methodology

We recorded vibrations from several points on server racks in an operative datacenter, using tri-axial accelerometers. These vibrations can be quantified on two different axes - the overall acceleration or total energy, called the *amplitude*, and the *frequency* or component frequencies of the signal. The amplitude is a scalar calculated from the power spectral density (PSD) function, or the root mean square value of multiple signals, in units of *grms*. The frequencies are in *Hertz*. We found that a rack server in an operative datacenter typically experiences vibrational frequencies ranging from 20 to 2000Hz, and amplitudes from 0 to 2 grms.

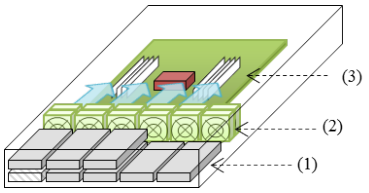


Fig. 1. Server organization with (1) hard disks and (2) fan assembly directing airflow towards (3) the motherboard.

Within the measured parameters, the vibrations are reproduced in a lab environment with an Unholtz-Dickie model K170 electrodynamic programmable vibrational table [47]. Table I lists detailed specifications of the platform and environment. The test server is mounted on top of the shake table, as we monitor the same points where it would have come in contact with a rack mount to ensure that the vibrations are faithfully transmitted. The test server has a commonly used single-socket, multi-core and multi-threaded processor, two memory sockets on either side of the processor, 6 fan modules, and 8 disk drive slots. It is loaded with a broad range of disk models as described in Table II, including commodity SATA drives, enterprise SAS drives, and solid state drives (SSDs). Since SSDs do not depend on moving parts to read data, they are impervious to vibrations. These control results are omitted for clarity. Fan speeds are controlled through pulse width modulation (PWM). This electrical "pulse" does not contribute to mechanical vibrations. The available fan speeds are 0-100% at increments of 10% (given some tachometer error) but in practice, fans are observed to be at least 50% when a server is active. We can temporarily override the built-in fan control algorithm to manually set fan speeds via the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE II
DISK DRIVE MODELS SPECIFICATIONS AND CHARACTERIZATION

Model	Type	Spin speed (RPM) *(specification)*	Max write speed (MB/s) *(measured)*	Write speed at max fan *(measured)*	Abbreviation
Fujitsu MHY2200BS	SATA	5400	31.2	6.2	FUJSATA
Hitachi Travelstar E5K500	SATA	5400	37.0	14.6	HITSATA
Seagate Savvio 10K.3 ST930003S	SAS	10000	72.2	72.2	SEASAS A,B
Hitachi Ultrastar C10K600	SAS	10000	81.6	81.6	HITSAS
Intel 710 SSDSA2BZ300G3	SSD	-	206.8	206.4	-

Intelligent Platform Management Interface (IPMI). To expose the true disk behavior, we disable the buffer cache that would have hidden disk access latency from the user. We run a pure I/O generator which issues random sustained writes to the disk, utilizing 100% of the I/O bus bandwidth. We quantify the effect of fan speeds on disk performance in terms of data write throughput. The top write speeds measured while the server is experiencing no external vibrations are reported in Table II. In a latter Section VI, we evaluate realistic database benchmarks with more variable I/O bandwidth requirements.

B. Amplitude test with random frequencies

We study the effect of external vibrations varying in *amplitude*, defined as the total combined signal strength of each component signal in the frequency profile. Vibrations are generated on the shake table while fan speeds are fixed at 50% PWM, and accelerometers are placed at rack-contact points on the server to report the total amplitude of vibrations delivered.We ran experiments with profiles that cover two different collections of frequencies. Figure 2 shows the effect of increasing vibration strength of frequencies ranging 20-800Hz, while Figure 3 shows the same for frequencies 20-2000Hz. Although lower throughputs follow higher amplitudes, the shape of the curve varies across hard drives and across frequency profiles. Of the two SATA drives spinning at the same speed (5400 RPM), FUJSATA performs better than HITSATA for $grms < 0.2$. At $grms = 0.63$, HITSATA writes at 8.6 MB/s in the first profile but 3 MB/s in the second. SAS drives are more resilient, showing signs of performance degradation at $grms = 1.27$. The largest drop among the SAS drives is 10.5% on HITSAS and the largest drop among the SATA drives when HITSATA stalls at 0 MB/s at $grms = 1.8$.

C. Frequency test with fixed amplitude

This experiment characterizes the hard disk response to external vibrations of varying frequencies. From on-site measurements at datacenters and observing Figures 2 and 3, we fixed the amplitude of vibrations at 0.17g, where drives experienced minor throughput degradation. Figure 4 shows the sweep through frequencies between 20 to 2000Hz and resulting disk throughput changes. Certain frequencies that cause performance degradation have a very narrow band. Even though more obvious degradation is seen at higher frequencies, there are narrow bands where disk performance returns close to its ideal. SATA drive throughput drops to 0MB/s at various points, while SAS drives fluctuate by 1-2%.

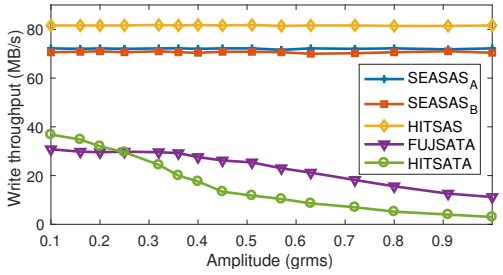


Fig. 2. Measured throughput dependence on vibrational amplitude, component frequencies ranging 20-800Hz

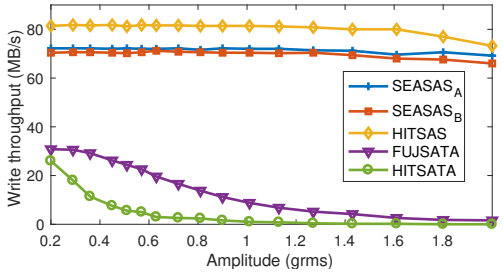


Fig. 3. Measured throughput dependence on vibrational amplitude, component frequencies ranging 20-2000Hz

D. Fan sweep test

Here, we isolate the effect of internal vibrations generated by the full range of possible fan speeds by bolting the server to the stationary shake table. With each change in stimuli, the disk drive throughputs take 20 seconds to respond. In our experience, the processor shuts down within 10 seconds of turning off the fans, while self-reporting on-die temperatures up to 91°C immediately before crashing. Consequently, it is challenging to accurately measure system characteristics in fine-grained steps at low fan speeds. We step through fan speeds from 100% to 0% PWM at 10% step sizes to obtain stable results. Figure 5 shows the average degradation of write throughput on fan speeds, normalized to the maximum throughput measured on each disk. There are no observable vibrational effects below 50% PWM. SATA drives show the most throughput degradation, down to 35% and 12% of their maximum value. The SAS drives are only affected by the highest fan setting - HITSAS loses 2% of its throughput.

With these experiments, we have characterized the relationship between hard disk performance and vibrations they experience. The possible effects of vibrations external to the server

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

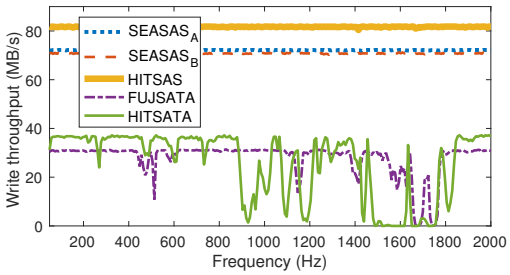


Fig. 4. Measured throughput dependence on vibrational frequency with amplitude fixed at 0.17g

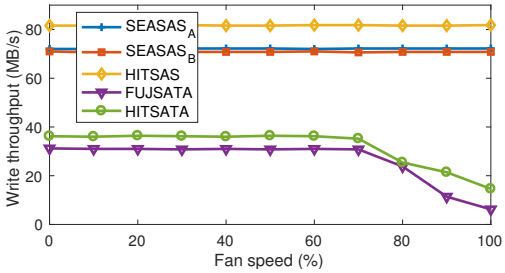


Fig. 5. Measured throughput dependence on fan speeds (no external vibrations)

(represented by the amplitude and frequency sweep tests) are not easily mitigated - short of expensive hardware rehaul. Although the mechanical study of these drive differences is out of the scope of this paper, we do observe that enterprise SAS drives are consistently more resilient than commodity SATA drives. Enterprise drives tend to have a heavier and more stable chassis, more expensive servos controlling the read/write head, better spindle motor shaft capturing, and better air flow control, all of which improve resilience against environmental vibrations [3]. Since our motivation is to find solutions for in-server hardware management, and the majority of deployed drives in cost-sensitive datacenters are commodity SATA, we choose to focus on the relationship between internal fans and SATA disk performance. Based on measurements presented here, this leaves a 65-88% drive performance gap that we hope to close with intelligent fan control policies.

IV. Performance, Thermal, & Energy Models

In this section, we model major physical and software interdependencies in the system, which will enable the optimal hardware manager in Section V. These models are derived from real measurements of the processor, hard drive, power and thermal sensors, fan cooling subsystems, and mechanical environment while running database workloads on the server described in Table I. Well-known power and thermal models are calibrated based on temperature and power measurements of our machine. We use measurements of disk performance from the previous section to model the effect each cooling decision has on IO throughput and system performance. All these components are combined to represent the behavior of a high-end server running typical database workloads. For example, disk performance impacts database application performance, which drives to CPU utilization, which dissipates some amount of dynamic power, which finally affects chip temperature. The temperature drives fan cooling response, which may generate vibrations which degrade disk performance. We model each of these interactions serially, to avoid double-counting the indirect effects of interconnected factors.

A. Workload representation

We chose the TPC-H benchmark suite to represent data-intensive workloads that commonly run in datacenters [15]. TPC-H is a decision-support benchmark consisting of 22 queries representing different business-oriented queries on large datasets. Depending on the size of the dataset, the entire suite can take on the order of hours or days to complete. The benchmark specification states that performance is defined by the query throughput (i.e. query-per-hour) for a fixed dataset, processor parallelism and memory size. For each individual query of a 40GB database size, with 4 parallel core threads allowed, and 128GB RAM, we calculate performance as the execution time required. We extract the model for each query using the single-user "power test" scenario, as opposed to the "throughput test" which represents a multi-user environment.

Other researchers have had success categorizing database workloads solely on their observed disk activity fluctuations, without tracking the semantics at an application level [19], [20]. Since different database operations in a single query can activate parallel cores, memory, and the IO bus in different patterns, we extend the model to represent a more comprehensive view of the system operating constraints, using a database manager that enables parallel queries where appropriate. We monitored the system using built-in trace commands (*mpstat, iostat, vmstat*) and a database monitor Oracle Enterprise Manager. Resource utilization can be described by vectors in a multi-core scenario in the form $< c_0, ..., c_{N-1}, io >$ where c_i represents the utilization between 0-100% for physical core i out of N cores, and io represents the percentage of maximum IO bandwidth (machine specification is 300MB/s).

We observe similarities among the observed utilization points and we model these similarities as system states. We use k-means clustering to quantify these similarities, where each cluster corresponds to a distinct system state. When determining a cluster for each observation, distortion is defined as the sum of the squared distances between each observation vector and its closest centroid [48]. The distortion decreases non-uniformly as the number of clusters increases. The elbow test described in [49] determines an appropriate number of clusters (k) to determine the point that gives the most benefit (in terms of reducing distortion) relative to an increase in clusters. Consider the decreased distortion per increment in k as the quantifiable benefit of increasing k. Then the first derivative represents the rate of gain in benefit. Furthermore, to find the k setting that yields a highest gain in benefit vs. increase in k (and consequently, lower benefit for $k+1$) we can take the derivative of the rate of gain. Thus, identifying the minimum in the second derivative shows us an appropriate k using the "elbow" test. The statistics used can be collected at runtime to update the model after initial deployment, to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

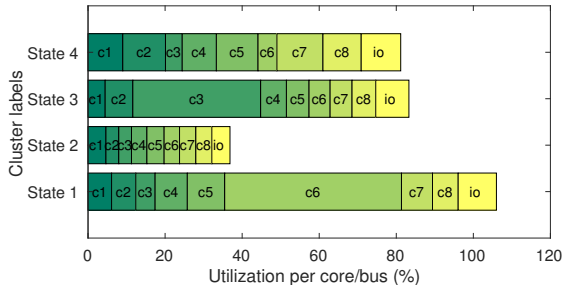


Fig. 6. Measured CPU utilization (c1-c8) and IO bandwidth (io) statistics serve as vector input into the k-means clustering algorithm. Each cluster then represents a distinct hardware state.

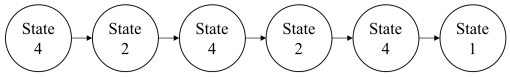


Fig. 7. Model representation of query 2, with a chain of 6 states

ensure that the workload clusters defined represent the range of resource utilization accurately.

With this method, we find that four clusters (Figure 6) provide a good tradeoff between number of clusters vs. distortion value. With all execution grouped into one of these four clusters, on average each query can be described with a chain of 97 states. Figure 7 illustrates how the shortest *query 2* is represented by a chain of 6 states. At the high level, most of the TPC-H queries consist of reading data from separate tables in parallel before sorting and/or joining them. Parallel operations show multiple cores being active - e.g. State 2 may be issuing multiple small read requests, while State 4 is issuing large bulk transfers. The joining and aggregation of parallel work present as one particular core being very active and others being relatively idle (e.g. State 1 and State 3).

The average length of time spent in each state per occurrence varies between states, though all are on the order of seconds. The performance traces were collected while the server was in a cool room, so the processor stayed cool and fans were running at low speed. This measured time is considered the "ideal" time since there are no vibration-induced delays. The duration of a state may be extended dynamically if the state is IO-dependent and fan speeds are high - as measurements show in the previous section.

In the rest of this work, we consider each query as a series of intervals, where each interval executes a single workload state. For example, the shortest *query 2* is represented with 6 states of various lengths, while the longest *query 1* has 330 state changes. The average query length across all 22 queries is 120 states where the standard deviation is 88 states. For each single workload state, since the hardware utilization patterns are constant, their power and thermal responses can be estimated. They are evaluated on an interval-by-interval basis to determine the necessary changes in cooling control.

B. Power model

In most server systems, including ours, the processor and the cooling system represent the majority of total server power consumption and have a wide dynamic range [50] [51]. The processor's high power density dominates dynamic changes in chip temperature at runtime. Processor power dissipation is further comprised of dynamic power - dependent on utilization state w - and static "leakage" power - dependent on chip temperature T. Thus power consumption ε_{power} is summarized as:

$$\varepsilon_{power}(w,T,f) = \varepsilon_{dynamic}(w) + \varepsilon_{static}(T) + \varepsilon_{fan}(f) \quad (1)$$

Our processor's maximum power dissipation by design (the "thermal design power") is 240W; we observe a typical range of 80-200W consumed by the processor, depending on utilization. We estimate the dynamic power dissipation of each workload state by linearly scaling the power range by the utilization factor [52]. Since the utilization level for each state is fixed as per the workload model, the dynamic power consumption is $\varepsilon_{dynamic}(w_i)$. For the typical range of operating temperatures in a server and time periods on the order of seconds, a linear model for static power has been shown to have an error less than 5% [53]. Thus, we approximate the static power $\varepsilon_{static}(T)$ as linearly dependent on temperature for a short workload interval. We use a cubic fan power model [6], calibrated by a reference constant r_f, based on a known fan speed f_r and its corresponding power consumption level p_r. Our fan power model is defined as: $\varepsilon_{fan}(f) = r_f f^3$, where $r_f = \frac{p_r}{(f_r)^3}$.

The fan speeds are reevaluted once per workload interval, on the order of several seconds. Thus, each interval power depends on the single workload state, the starting temperature, and the fan speed.

C. Thermal model

The major heat producing and extracting components in our server can be represented with electrical analogies [42] [45]. Components that consume power (such as the processing cores and caches) behave as heat sources, and are modeled as power sources in the circuit. The heat sink's heat dissipation is represented by convective resistance, which changes during runtime according to airflow volume. This air flow rate increases linearly with fan speeds [6].

We extract an analytical model based on initial RC simulations run on the widely accepted HotSpot tool [42]. The goal is to identify a differentiable model for a formal problem formulation in Section V. We simplify the model to focus only on runtime variations of the hottest core and the fan cooling capacity. The hardware has a thermal response time constant τ that is dependent on fan speed. Each workload has a time-invariant steady state temperature T_{ss} if it is allowed to run indefinitely at a fixed fan speed. Thus, beginning at some initial temperature T_0, for any single workload executing for a certain amount of time t, a cooling-dependent time constant τ dictates how quickly the system approaches the steady state temperature T_{ss}. The heat sink temperature falls linearly with the convective resistance - thus, temperature T_i decays as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

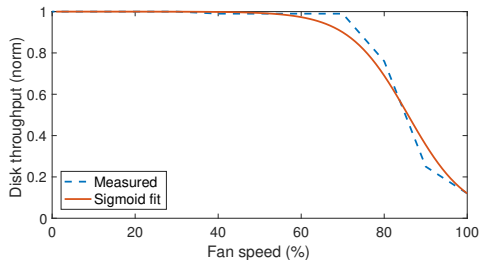


Fig. 8. Fan speed-disk throughput interaction fit to a sigmoid function

an exponential function of the given fan speed f_i. Recall that the actual length of each interval t varies depending on which workload state is executing, since we take into account both the nominal interval length of that particular state and any delay that the fan may incur. Equation 2 describes the instantaneous temperature at the end of interval i (i.e. the start of $i+1$) as:

$$T_{i+1} = T_i e^{\frac{-t}{\tau}} + T_{ss}(1 - e^{\frac{-t}{\tau}}) \tag{2}$$

The *initial* temperature of any given interval i is a historical (fixed) value T_{i-1} from the perspective of that interval. The *final* temperature of that interval T_i is calculated according to the initial temperature, the dynamic power dissipated by the current workload state, leakage power dissipation due to the starting temperature, and the cooling capacity of some chosen fan speed. This becomes the *initial* temperature of the next interval $i+1$.

We compared 18 seconds of time series temperature data from our analytical model and a full HotSpot simulation. Across the range of our expected workloads, the average error is less than 1%; thus we can approximate HotSpot's accepted model quite closely, with simulation time on the order of minutes instead of hours.

D. Cooling-vs-disk interaction model

In typical enterprise servers, vibrations are transmitted from the fan motors by mechanical coupling to the housing for the disk drives. In earlier work, an empirical curve was used in [41] without making assumptions about the exact relation between fan speeds and vibrational amplitudes. Equation 3 formalizes the relationship as a sigmoid function of fan speed f, also visualized in Figure 8. We model the Fujitsu SATA drives in our particular server with $\alpha = 1034.65$, $\beta = 1033.65$, $\gamma = 8.04$, for an R^2 value of 0.98 and average relative error of 2.7%. The optimal solution (Section V) depends heavily on the exact relation between a particular disk and the server's cooling system - the cost of inaccurate modeling for a particular device are discussed and quantified in Section VI-C. In an enterprise scenario, many of the same hardware models would be deployed at once, which should alleviate the initial cost of accurately characterizing this disk performance dependency.

$$ThroughputFactor(f) = \frac{\alpha}{\beta + e^{\gamma \cdot f}} \tag{3}$$

The workload model already contains information about the ideal runtime $c(w)$ of each workload state w, assuming full availability of the disk bandwidth (Section IV-A). It has been shown that throughput degradation has a superlinear effect that cascades into the overall application delay [12] [13]. In lieu of modeling memory hierarchy and database storage structures in detail, we assume that the final delay caused by fan degradation is at least inversely proportional to the throughput. The resulting execution time $\varepsilon_{time}(w, f)$ needed for executing a single instance of a workload state at a certain fan speed is then defined as:

$$\varepsilon_{time}(w, f) = c(w) \cdot \frac{\beta + e^{\gamma \cdot f}}{\alpha} \tag{4}$$

V. Optimal cooling control

We consider a formal constrained optimization problem to define fan speeds that minimize the total cost of system operation. Each database workload is represented as a chain of workload states, where a workload state w_i identifies the system resource vector during some execution interval i. Each interval has a cost of execution, ε_{cost}, and the total cost C_N of executing N intervals is the sum of each interval cost. We minimize the cost while keeping the temperature of all components under the threshold T_{limit} at all times:

$$\min_f [\boldsymbol{C_N}(f)] \text{ s.t. } T_i \leq T_{limit} \forall i \in [0, N] \tag{5}$$

To minimize the energy cost of execution, Equation 5 would be rewritten with $C_N = E_N$, where the energy consumption E_N is an accumulation of power consumption values ε_{power} scaled by the interval lengths of ε_{time}.

$$E_N = \sum_{i=1}^{N} \varepsilon_{power}(w_i, T_{i-1}, f_i) \cdot \varepsilon_{time}(w_i, f_i) \tag{6}$$

Slater's condition states that any feasible solution to the Lagrangian Dual problem is also an optimal solution for a convex objective function [54]. Since the constraint function is the temperature, it suffices to analyze the various power components along with the time degradation dependency for convexity.

Lemma V.1. *Processor and fan power consumption power are convex with respect to fan speed.*

Proof. Power consumption consists of three components: 1) processor dynamic power, a linear function of processor utilization; 2) processor static power, a linear function of temperature; and 3) fan power, a cubic polynomial with respect to fan speed. Processor dynamic power is only a function of utilization, thus unaffected by the fan speed selection: $\frac{\partial^n \varepsilon_{dynamic}}{\partial f^n} = 0, \forall n$. Processor static power has a linear relation with temperature, thus the convexity of static power is the same as of temperature. Consider the temperature function in Equation (2). The steady state temperature (T_{ss}) and time constants (τ) are empirical values obtained from physical characterization, so we prove the convexity of temperature through numeric differentiation in Figure 9. The second derivative is non-negative for all fan speeds, so temperature as a function of fan speed is convex. The power consumption of the fan is a cubic polynomial with positive first and second derivatives:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

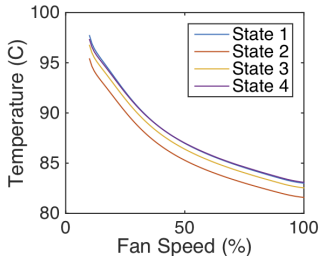


Fig. 9. Numerical function of temperature with respect to fan speeds

$\frac{\partial^2 \varepsilon_{fan}(f_i)}{\partial f_i^2} = 6 r_f f_i > 0$. Thus all power functions are convex with respect to fan speeds. □

Lemma V.2. *Execution delay is convex with respect to fan speed.*

Proof. Total execution time of any given interval is a function of the workload being executed, which has a minimum delay ($c(w)$), and the fan speed, which may further slow down the workload. The delay function (Equation 4) is an exponential, and its second derivative is: $\frac{\partial^2 \varepsilon_{time}(w,f_i)}{\partial f_i^2} = \frac{c(w)\gamma^2}{\alpha} e^{\gamma f_i} > 0$. Since α, β, and $gamma$ are all positive constants, the second derivative is always positive, hence the delay function is convex with respect to fan speeds. □

Theorem V.3. *Total energy cost of any application executed on this hardware platform is convex with respect to fan speeds.*

Proof. Energy consumption is defined as the product of power consumption and the total time that power is dissipated across. The first derivative of the energy cost function is always positive, such that energy monotonically increases with fan speeds. The final convexity of energy is calculated as: $\frac{\partial^2 E_i}{\partial f_i^2} = \frac{\partial^2 \varepsilon_{power}(f_i)}{\partial f_i^2} + \frac{\partial^2 \varepsilon_{time}(f_i)}{\partial f_i^2} + 2\frac{\partial \varepsilon_{power}(f_i)}{\partial f_i}\frac{\partial \varepsilon_{time}(f_i)}{\partial f_i}$ Moreover, since we find that the second derivative of the energy cost function with respect to fan speeds is always positive, the problem is convex. □

A. Convex optimal formulation

We minimize the cost C_N while ensuring that the system remains strictly *below* the temperature constraints (T_{limit}) for all time intervals. The Lagrangian with KKT multipliers is formulated as such, where each λ_i represents the constraint at interval i:

$$\mathcal{L} = C_N + \sum_{i=1}^{N} (T_i - T_{limit})\lambda_i \tag{7}$$

We need to solve for the fan assignment f at every interval j such that the Lagrangian is minimized.

$$\frac{\partial \mathcal{L}}{\partial f_j} = \frac{\partial}{\partial f_j}(C_N + \sum_{i=1}^{N} (T_i - T_{limit})\lambda_i) = 0, \forall j \in [1, N] \tag{8}$$

The total cost C_N is a summation of all interval costs, and is dependent on all fan speeds. We assume that each interval's fan mainly affects its own interval cost, and less so intervals before or after it. This means dropping the derivative of the static power term (ε_{static}), since it is the only term that carries the hysteresis in terms of fan-dependent temperature. That is, the dominant dependency of total cost C_N on f_j is the the cost of that interval $\varepsilon_{cost}(w_j, T_j, f_j)$. Thus, $\frac{\partial C_N}{\partial f_j}$ simplifies to $\frac{\partial \varepsilon_{cost\,j}}{\partial f_j}$. Additionally, since temperatures in the past are not dependent on current or future fan settings, the summation begins at the relevant interval j instead of 1.

Based on the temperature model in Equation 2, the derivative of the change in temperature ΔT converges to $\frac{\partial \Delta T}{\partial f} = \frac{\partial T_{ss}(1 - e^{\frac{-t}{\tau}})}{\partial f}$. For consecutive intervals j and $j+1$, we divide the Lagrangian minimizations by $\frac{\partial \Delta T_j}{\partial f_j}$ and $\frac{\partial \Delta T_{j+1}}{\partial f_{j+1}}$ respectively, then take the difference. Since temperatures must stay strictly within constraints, the KKT multiplier λ_j must be equal to 0. The expanded summations simplify to this equality:

$$\boxed{\frac{\frac{\partial \varepsilon_{cost\,j}}{\partial f_j}}{\frac{\partial \Delta T_j}{\partial f_j}} - \frac{\frac{\partial \varepsilon_{cost\,j+1}}{\partial f_{j+1}}}{\frac{\partial \Delta T_{j+1}}{\partial f_{j+1}}} = 0} \tag{9}$$

Intuitively, this specifies that the ratio between *execution cost* and the *thermal pressure* should be held constant across intervals. This simplifies the Lagrangian problem into finding a fan setting where this ratio can be kept constant throughout runtime.

B. Optimal algorithm design

We use *energy* as an example of an optimization objective in the rest of this paper. The proposed algorithm 1) searches for a convex optimization of energy costs, 2) while guaranteeing that temperature constraints are met, 3) for a set of known queries on a known system. Such a solution may not exist, hence necessitating a search. Figure 10 shows the logic flow of iterating over each query to solve for a vector of optimal fan speeds for a given workload and initial temperature. A more detailed pseudocode for the interior point search for a solution is described in Algorithm 1. In each workload interval, the controller takes the *current workload state* and a *target ratio* as inputs, and solves the combined system model to output the closest permissible *fan speed* that produces a matching ratio, to fulfill Equation 9. In lieu of physical sensors, we use Equations 1 and 2 to represent power and thermal interactions. Equation 4 dictates the effect a fan setting has on the execution time of each interval.

To begin, we start the query at the lowest possible fan speed (line 1). This first interval drives a target ratio for all following intervals (lines 2-5). According to Equation 9, this is a potential value of the initial ratio, $ratio_0$, that should be matched for the rest of execution in order to achieve the minimal cost of execution, or *minimal energy* in our case. The rest of the simulation (power dissipation and temperature simulation) follows this decision. For all following intervals, there are fixed costs that are independent of the fan decision, including dynamic power dissipation and static power dissipation (lines 7-8), after which the solver attempts to set a fan speed that matches $ratio_0$ as closely as possible (line 9). After making the interval decision, the solver completes timing and thermal modeling (lines 11-12).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

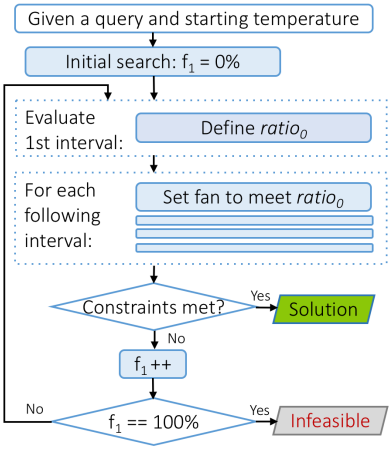


Fig. 10. Control flowchart of search algorithm for a known query.

Due to physical limitations of the fan speed and a possibility of overloading the processing workload, there may not be any feasible fan speeds that satisfy $ratio_0$ without violating temperature constraints. If the constraints are met at the end of a query run (line 17), the corresponding lowest-cost fan assignment is returned as an optimal solution. If one of the constraints is violated, the workload chain is re-evaluated, starting with the next lowest possible fan speed (loop to line 1). If all fan speed have been exhausted and there is still no solution that meets all constraints, that means the the problem is infeasible, and the only resort is to slow down the CPU workload with DVFS.

Since the total cost always increases with fan speeds, the algorithm is described linearly here for clarity, but can be sped up by doing a binary search. Pragmatically, the number of system states and quantized fan speeds are both limited (e.g. only 10 fan settings in our actual server); these values can be pre-computed and stored in a lookup table for execution at runtime.

VI. Results

In this section, we demonstrate the effectiveness of our proposed solution by comparing against the state of the art and other proposed solutions in literature. We describe the hardware and software setup of our physical measurements on the real server, as well as the parameters of our modeling and simulation. We describe three state of the art fan control strategies and compare with our results. Finally, we discuss the robustness of our modeling and optimal solver, analyzing how optimization results might change at various levels of model inaccuracies.

A. Experimental Setup

We model the same server instrumented and measured in Section III, a SPARC T4-1 server with 8 cores running at 2.85GHz, with 8 DIMM modules of 16GB each. We use commodity SATA disks as they are preferred by cost-sensitive datacenters for their low cost per storage density. Buffer caches are enabled to capture the real response of applications along with power, thermal, cooling and disk performance issues.

Algorithm 1 Search for energy-optimal fan speeds

1: **for** f_0 = each increasing fan setting **do**
2: **for** the first interval **do**
3: $ratio_0 \Leftarrow$ given f_0, find $\partial\varepsilon_{energy}/\partial\Delta T$
4: $t_i \Leftarrow$ given $\{workload, t_0, f_0\}$, find temperature
5: **end for**
6: **for** each following interval i: **do**
7: $dynamicPower \Leftarrow$ fixed for the $workload$
8: $staticPower \Leftarrow$ fixed for t_i
9: $f_i \Leftarrow$ find fan to match $ratio_0$
10: $fanPower \Leftarrow$ calculate fan power
11: $intervalTime \Leftarrow$ find fan-induced delay
12: $t_{i+1} \Leftarrow$ advance temperature
13: **if** t_{i+1} violates constraints **then**
14: continue to next f_0
15: **end if**
16: **end for**
17: **if** workload completes within constraints **then**
18: minimum cost fan assignment found!
19: return solution $\bar{f}$
20: **end if**
21: **end for**

We evaluate the management policies with a mixed workload of database queries and compute-intensive batch jobs as per typical datacenter environments. TPC-H is a decision support benchmark representing databases requests [15]. The queries combine operations such as sequential scan, index scan, merge join, and hashing functions. The thermal threshold is set to 85°C. SPEC CPU 2006 is a benchmark suite targeted towards compute-intensive workloads [55]. We assume there are four co-located compute tasks on the processor, represented in our power and thermal simulations as single-threaded tasks that consume 8W each - this number was obtained from averaging the power consumption of SPEC CPU 2006 benchmarks. With this mixed workload on our physical system, we encounter both thermal issues due to heavy computation, and I/O performance issues due to reliance on the disk access rates.

Figure 11 summarizes the flow of data from physical measurement to simulation. We monitor sensor statistics and event logs on a real server through IPMI [56]. Most enterprise servers have a side-band controller implementing IPMI to handle server management, reading hardware sensors, and enforcing power modes. Disk access statistics are collected through *iostat* reports, estimating the number and average service times of queued and active transactions per sampling interval (every second). These event logs are converted into a discrete workload model as described in Section IV. The advancement of workoad states varies, measured at a granularity of 10ms. Since the packaging thermal time constant is at the order of seconds, the fan control interval is set to 1s. In practice, the side-band controller executes the fan control algorithm. Unfortunately, we were unable to implement custom control policies on the machine due to permission restrictions on user programmability. Thus, we use MATLAB to compare multiple algorithms by replaying and manipulating

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

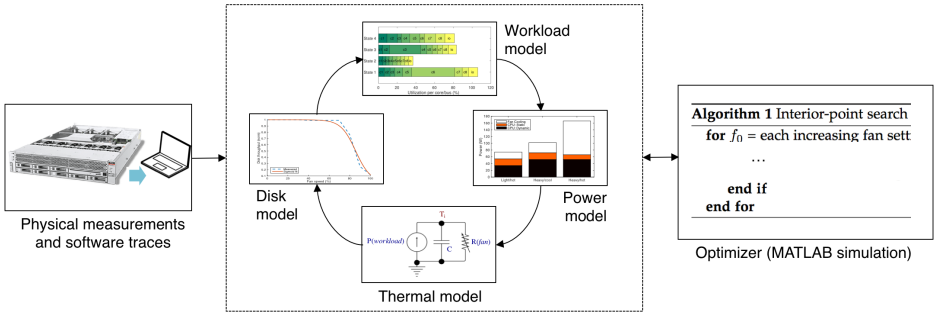


Fig. 11. Subsystem models are based on real physical measurements. We use MATLAB to coordinate the models and optimize the system.

our physical experimental traces and analytical models.

We compare our proposed controller against three others in terms of delay and energy cost when running mixed SPEC and TPCH workloads. They represent a range of sophistication and complexity in hardware management schemes. The first (PID [36]) is time-tested strategy used in many control systems in various engineering fields, representing the state of the art. The next two strategies (Adaptive PID [37] and JETC [44]) were proposed in literature; they make cooling decisions by accounting for temperature conditions as well as CPU performance degradation. If any controller fails to maintain chip temperatures under the specified threshold (85°C in our system), the hardware enters an emergency state, where chip hardware enforces progressive power gating to throttle performance and reduce power dissipation relative to the magnitude of temperature violation. Individual policy details are described below:

Proportional-Integral-Derivative (PID) [36] is completely agnostic to workload and reacts only to temperature sensor feedback. Being a reactive method, it responds much slower to temperature fluctuations than proactive controllers, and by nature allows both over- and under-corrections before arriving at a steady solution. We show results for "PID-1" which is the same strategy with a setpoint conservatively set below the threshold (by 1°C in our case) to reduce temperature violations. The tuning parameters for both PID and PID-1 are determined using the Ziegler-Nichols closed loop tuning method [57]. The control interval is set to 10 seconds.

Adaptive PID [37] refines the PID assignment into two zones and scales the tuning parameters dynamically based on the current fan region. It uses the Ziegler-Nichols closed-loop tuning method [57] to obtain PID parameters specific to a high and low fan setting (15% and 65% of the maximum, in our experiments). For all fan speeds between those two settings, the paramters are linearly interpolated for faster convergence. The original proposal for APID stated a control interval of 30s, aiming to converge the fan control within hundreds of seconds. In our experience, a maximum control interval of 10s is required to maintain steady chip temperatures.

Joint Energy, Temperature and Cooling Manager (JETC) [44] uses proactive core migration to control heat generation. In each control interval, this policy predicts the upcoming power dissipation and resulting temperatures. Using It then uses an RC thermal model proposed in [42] to calculate the required increase or decrease in cooling capacity to bring temperatures to the system thermal setpoint. As in the source paper, JETC re-evalutes control decisions every second, predicts temperatures at a 9 second horizon. While making these decisions, the fan controller aims to minimize fan setting changes during runtime.

Our **Energy-Optimal** controller implements the search described in Algorithm 1, minimizing for total energy consumption. The search is executed offline, then applied to a known query at runtime. For each workload state in a query, it sets the fan to maintain a constant ratio between the change in energy and the change in temperatures. Unlike other policies, control decisions are made when the workload state changes instead of a fixed control interval. In practice, the control interval is on the order of seconds.

B. Controller policy results comparison

The energy and delay results from a select number of TPC-H queries are shown in Figure 12. By design, the energy-optimal solver yields fan settings that yield the lowest possible server energy consumption required to complete a workload. Our energy savings come predominantly from faster completion times and lower overall fan speeds. This controller operates as close to the temperature threshold as possible without crossing it, unlike the oscillatory nature of PID solutions [36]. Our policy guarantees zero temperature violations with lowest energy consumption possible. Only the PID-1 controller is able to keep temperatures below the threshold, at a much higher cost in terms of delay and energy since it sets an artificially lower thermal setpoint. The slow convergence of APID leads the policy to violate temperature constraints. On average across queries, our policy is 2x faster than these heuristic solutions, while consuming 65% less energy. JETC relies on a detailed thermal simulation of the processor package. At each decision interval, it predicts the upcoming temperatures and power dissipation, then calculates the heat sink cooling capacity needed for such a power density to maintain temperatures under specified constraints. For a slowly-varying workload, this yields stable temperatures and converges to a solution much faster than the PID-based controls. However, because JETC relies on migration-based management, its actually results in unstable temperatures for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

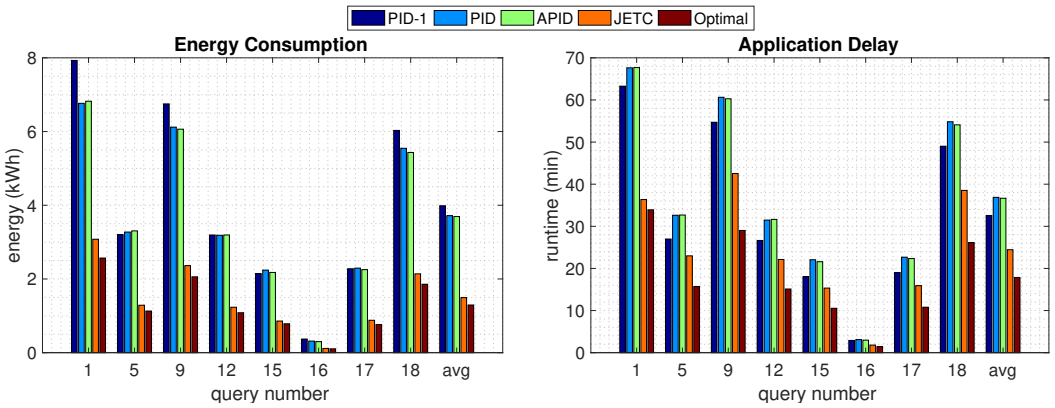


Fig. 12. Final simulation results of cooling policies on TPC-H queries. The selected queries represent various lengths and operations across the suite.

TABLE III
FAN CONTROL BEHAVIOR AVERAGED ACROSS SELECT QUERIES

Algorithm Name	PID [36]	APID [37]	JETC [44]	Optimal
Average fan speed (% of max)	77	58	63	57
Std dev of fans (%)	10.2	12.8	6.6	20.1
Time in emergency (%)	28.6	32.1	12.1	0
Avg. temperature (°C)	84.9	74.1	84.2	84.9
Peak temperature (°C)	91.5	91.2	85.5	85.0

our workloads (this effect was also noted in [37]). To compensate for these unstable temperatures, this control requires higher fan speeds on average. Compared to JETC, our optimal policy is 1.5 faster on average, using 19% less energy.

Table III summarizes controller behavior in terms of fan speeds and resulting temperatures over all tested queries. By design, the optimal controller never exceeds the temperature threshold, while the PID-based solutions naturally overshoot the temperature setpoint regularly. JETC also makes incremental steps towards the steady state fan speed, with possible temperature violations while in progress. Such temperature violations incur throttling delays due to hardware-enforced power gating (listed as "Time in emergency" in the table), while the total server energy consumption continues to rise. Meanwhile, the optimal solution finds the minimal fan speeds to keep temperatures within limits immediately without need for oscillation like the PID-based solutions, resulting in a single change in fan settings following each workload state change. Our setup (25°C ambient and 30°C server internal) already pushes the limits of operating range recommended by the vendor for running high-performance mixed workloads. Additional thermal pressure from the ambient environment can be simulated via the steady state temperature (T_{ss}) of each workload state. At 5°C hotter, there is no feasible solution that exists without resorting to some time in emergency mode. At 5°C cooler, our policy still delivers the best energy results, but the magnitude of benefit over the most energy-expensive PID controller drops from 65% to 36%.

C. Sensitivity to model inaccuracy

Our optimal solver relies on several models to represent physical subsystems in the server. Although we were unable to implement and evaluate the accuracy of our final algorithm in a real machine, we present some studies to evaluate the effectiveness of the convex optimal formulation when component models may be inaccurate.

1) Power model accuracy: Errors in the power model are described with additive Gaussian white noise in Equation 1 used by the solver at each interval. The signal-to-noise ratio (SNR) is shown in decibels (dB), where a higher SNR represents a "clearer" original signal relative to the noise. Figure 13 shows that a noisy power model results in sub-optimal results and higher energy cost, though the optimal policy still outperforms the next-best JETC policy for most queries. An SNR of 20dB indicates that original signal is 100 times more powerful than the noise. Due to noise added to the power model, the optimal policy consumes on average 6.3% more energy as compared to the optimal policy that uses power model with no noise added. JETC consumes 8.5% more energy when noise is added to the power model as compared to no noise. Heuristic PID and APID policies are not sensitive to the noise in the power model, as they respond only to temperature readings, so their results remain the same regardless of the model. Thus, optimal policys benefit relative to PID and APID policies is reduced on average from 65% to 63% with power model of 20dB SNR as compared to when model with no noise is used.

2) Thermal model accuracy: We investigate the effects of error in the temperature model (T_{ss} in Equation 2). We again use additive Gaussian white noise to estimate errors in the thermal model. Figure 14 shows that due to a 20dB SNR in the temperature model (i.e. noise is 1% of the original signal), the optimal policy consumes 6% more energy as compared to when there is no noise in the model. It still outperforms PID by 63%. JETC has a much higher penalty due to noise in the thermal model - its energy cost increases by 95%, consuming 2x as much energy as the optimal policy at the same noise level. We conclude that while both the optimal policy and JETC depend strongly on a temperature model to achieve their objectives, the optimal policy is more robust to reasonable

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

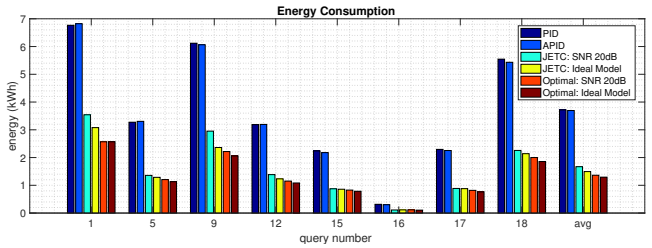


Fig. 13. Optimal solver efficacy under inaccurate power modeling

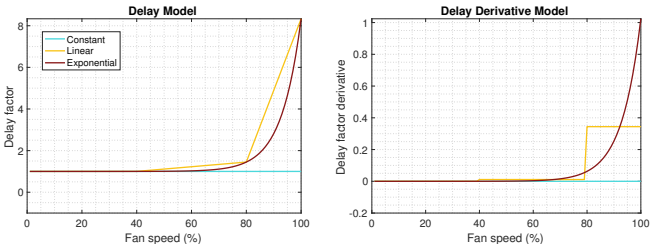


Fig. 15. Alternative disk delay models used by the optimal solver

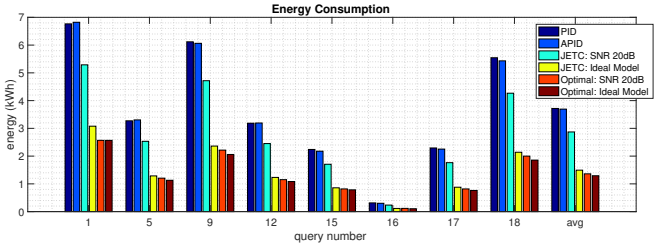


Fig. 14. Optimal solver efficacy under inaccurate temperature prediction

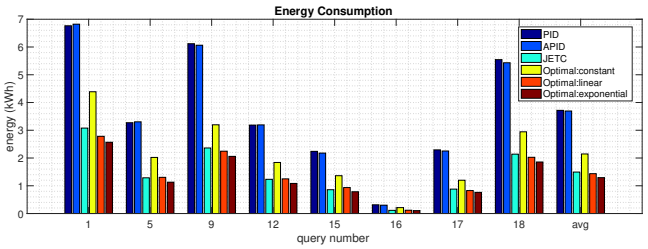


Fig. 16. Efficacy of the optimal solver with inaccurate delay models

levels of added noise.

3) Disk delay model accuracy: A major motivation of this work was to recognize that fan speeds have a detrimental effect on disk peformance [12], [41]. The original delay model we proposed in Equation 4 is exponential with respect to fan speeds. Figure 15 illustrates two alternative models: a simple piecewise linear function and a constant function, representing the common assumption that fan speed does not affect performance, as well as their first derivatives which dominate the convex optimization solution in Equation 9. To test these results, we substitute each of these alternative models into the optimal solver calculation, and simulate the final results for a system that still responds with an exponential disk delay relative to fan speeds. Figure 16 shows that if the solver tries to obtain an energy-optimal solution while assuming that disk throughput is independent of fan speeds, it uses 31% more time and 71% more energy to complete the workload on average. A piecewise linear model has a delay penalty of 4% compared to an accurate exponential model, using 12% more energy. Model-predictive controllers are meant to rely on reasonably assumptions about the physical system. Since JETC compensates for modeling errors through dynamic reactions, it can perform better than our entirely model-dependent algorithm in some cases. However, our formulation with a simplistic model still outperforms other algorithms on average, reducing system energy by 37% compared to PID.

D. Overhead and repeatability

Table IV compares different sources of overhead for each policy. "Design effort" represents the *offline* time and effort required to manually define or calibrate a controller before a query can be managed during real execution. Iteratively tuning a PID controller [36] takes many multiples of the machine's thermal time constant - minutes per iteration, and the Adaptive PID controller has to be tuned twice [37]. In contrast, both JETC [44] and our strategy require up front modeling effort to capture system physical characteristics. Our solver's workload trace clustering implemented in Python took less than 10 seconds over thousands of samples collected over 4.5 hours of actual database execution. In addition, we perform a search at design time for fan settings given starting temperatures and a known workload. MATLAB solves the longest query in 3.7 minutes. "Decision delay" is the *online* delay to process inputs and produce a decision at each control interval. Since we did not implement the algorithms in a real machine, these numbers are estimated from MATLAB execution times. For PID and APID, the controller computes the derivative and integral of historical errors, which contributes to the delay of over 7μs for each decision, on top of their intrinsic reaction delays to sensor stimuli. JETC simulates the processor floorplan in detail to decide fan speed, which takes on average 4.14 μs. Our runtime overhead costs (1.83 μs per decision) come only from accessing hardware sensors and looking up the precomputed solution. Our modeling methodology and fan control design is meant to be repeatable across other server configurations. The fan algorithm should always yield energy-optimal results if these assumptions hold: (1) The workload is predictable such that hardware utilization can be identified as distinct phases in a sequence. Our *a priori* solution is not applicable to workloads with very bursty or unpredictable hardware patterns. (2) Measured relationship between fan speed and disk delay is convex. While this has been observed in other cases [12], it may not hold true for all server configurations. (3) Server administration ensures that workloads are not grossly oversubscribed. Most enterprise servers are expected to operate far under 70% utilization on average, but if all cores are 100% utilized and/or overclocked for a period, temperature violations may be unavoidable. The technique may be modified with loosened constraints in exchange for some time delays, in order to settle for a near-optimal fan assignment.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCAD.2018.2855122, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems

TABLE IV
ESTIMATED OVERHEAD OF FAN CONTROL STRATEGIES

Algorithm	PID [36]	APID [37]	JETC [44]	Optimal
Requirements	Tuning	Tuning	Modeling	Modeling, solution search
Design Effort	Minutes	Minutes	Hours	Days
Decision Delay (each)	7.81 μs	7.89 μs	4.14 μs	1.83 μs

VII. CONCLUSION

In this work, we present the very first optimal fan control policy based on modeling a complex enterprise server platform, including hardware and data performance interactions. We identify a previously neglected link between the cooling subsystem and application performance, where commodity drive sensitivity to fans can cause up to 88% lower performance in realistic cooling situations. This sensitivity can further degrade the perforamnce of data-intensive workloads, resulting in wasted energy consumption. We also develop and integrate interdependent analytical models for performance, power, thermal, cooling. Finally, we define a multi-model objective function that can be solved to find optimally low-cost fan speeds, saving 19-65% of CPU and fan energy while guaranteeing that critical thermal constraints are still met 100% of the time. Where physical enclosure designs have failed to completely solve the problem, our method can be delivered via firmware updates to help the system relearn hardware states according to new physical configurations and upgrades (e.g. adding memory modules, upgrading storage drives). It also allows for fast and low-cost deployment.

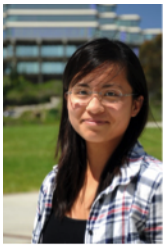
ACKNOWLEDGMENT

This work was supported by funds from Semiconductor Research Corporation (SRC), including CRISP, one of six centers in JUMP sponsored by DARPA, Terraswarm Research Center, Global Research Collaboration Task 2169, and Multi-Scale Systems Research Center. It was also supported in part by the UCSD Center for Networked Systems, NSF grants 821155, 916127, 1029783, 1730158, and 1527034, Oracle Labs, Google, and Microsoft.

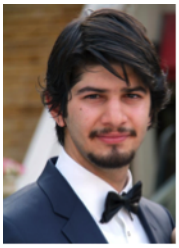
REFERENCES

[1] H. H. Liu, *Software performance and scalability: a quantitative approach*. John Wiley & Sons, 2011, vol. 7.
[2] F. Moore, "Tiered storage takes center stage," *Horison Information Strategies*, vol. 22, 2011.
[3] D. Anderson, J. Dykes, and E. Riedel, "More than an interface-SCSI vs. ATA." in *FAST*, vol. 2, 2003, p. 3.
[4] J. Ousterhout, A. Gopalan, A. Gupta, A. Kejriwal, C. Lee, B. Montazeri, D. Ongaro, S. J. Park, H. Qin, M. Rosenblum *et al.*, "The ramcloud storage system," *ACM Transactions on Computer Systems (TOCS)*, 2015.
[5] A. Capozzoli and G. Primiceri, "Cooling systems in data centers: state of art and emerging technologies," *Energy Procedia*, 2015.
[6] M. K. Patterson, "The effect of data center temperature on energy efficiency," in *The 11th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. IEEE, 2008.
[7] R. F. Sullivan, "Alternating cold and hot aisles provides more reliable cooling for server farms," *Uptime Institute*, 2000.
[8] R. Ayoub, S. Sharifi, and T. S. Rosing, "Gentlecool: Cooling aware proactive workload scheduling in multi-machine systems," in *Proc. Conference on Design, Automation & Test in Europe (DATE)*. IEEE, 2010, pp. 295–298.
[9] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, "Making scheduling 'cool': Temperature-aware workload placement in data centers," in *USENIX Annual Technical Conference, General Track*, 2005.
[10] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis Lectures on Computer Architecture*, 2013.
[11] K. Vaidyanathan, K. Gross, and S. Sondur, "Ambient temperature optimization for enterprise servers: Key to large-scale energy savings," in *Fourth Berkeley Symposium on Energy Efficient Electronic Systems (E3S), 2015*. IEEE, 2015, pp. 1–3.
[12] C. S. Chan, Y. Jin, Y.-K. Wu, K. Gross, K. Vaidyanathan, and T. Rosing, "Fan-speed-aware scheduling of data intensive jobs," in *Proc. ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*, 2012, pp. 409–414.
[13] J. Turner, "Effects of data center vibration on compute system performance," in *Proc. USENIX Conference on Sustainable Information Technology*. USENIX Association, 2010.
[14] P. Bodik, M. Goldszmidt, A. Fox, D. B. Woodard, and H. Andersen, "Fingerprinting the datacenter: automated classification of performance crises," in *Proc. European Conference on Computer Systems*. ACM, 2010, pp. 111–124.
[15] TPCH, "TPC-H Benchmark Suite," 2011. [Online]. Available: http://www.tpc.org/tpch/
[16] M. Akdere, U. Çetintemel, M. Riondato, E. Upfal, and S. B. Zdonik, "Learning-based query performance modeling and prediction," in *IEEE 28th International Conference on Data Engineering (ICDE)*. IEEE, 2012.
[17] A. Ganapathi, H. Kuno, U. Dayal, J. L. Wiener, A. Fox, M. Jordan, and D. Patterson, "Predicting multiple metrics for queries: Better decisions enabled by machine learning," in *IEEE 25th International Conference on Data Engineering (ICDE)*. IEEE, 2009, pp. 592–603.
[18] H. Jung, P. Rong, and M. Pedram, "Stochastic modeling of a thermally-managed multi-core system," in *Proc. Annual Design Automation Conference (DAC)*. ACM, 2008, pp. 728–733.
[19] S. Sankar and K. Vaid, "Storage characterization for unstructured data in online services applications," in *IEEE International Symposium on Workload Characterization*. IEEE, 2009, pp. 148–157.
[20] C. Delimitrou, S. Sankar, B. Khessib, K. Vaid, and C. Kozyrakis, "Time and cost-efficient modeling and generation of large-scale tpcc/tpce/tpch workloads," in *Topics in Performance Evaluation, Measurement and Characterization*. Springer, 2012, pp. 146–162.
[21] T. M. Ruwart and Y. Lu, "Performance impact of external vibration on consumer-grade and enterprise-class disk drives," in *Proc. IEEE/NASA Goddard Conference on Mass Storage Systems and Technologies*. IEEE, 2005, pp. 307–315.
[22] J. Yang, C. P. Tan, Z. He, Z. Y. Ching, and C. C. Tan, "An effective system-level vibration prediction analysis approach for data storage system chassis," *Microsystem Technologies*, 2016.
[23] D. Anderson, K. Green, O. Herrera, K. Schneebeli, and P. Urbisci, "Disk-drive chassis for reducing transmission of vibrations between disk-drive units of a disk-drive array," US Patent 6,154,361, 11 28, 2000.
[24] X. Tan, B. G. Chen, B. J. Zhang, B. C. Liu, B. N. Ahuja, I. J. Zhang *et al.*, "An advanced rack server system design for rotational vibration (RV) performance," in *15th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. IEEE, 2016, pp. 1320–1325.
[25] Y. Uzuka, K. Itakura, N. Yamaoka, and T. Kitamura, "Power supply, cooling and mechanical technologies for green it," *Fujitsu Sci. Tech. J*, vol. 47, no. 2, pp. 157–163, 2011.
[26] M. A. Bell, "Use best practices to design data center facilities," *Gartner Research. April*, vol. 22, 2005.
[27] S. Legtchenko, X. Li, A. I. Rowstron, A. Donnelly, and R. Black, "Flamingo: Enabling evolvable HDD-based near-line storage." in *FAST*, 2016, pp. 213–226.
[28] S. Sankar, M. Shaw, K. Vaid, and S. Gurumurthi, "Datacenter scale evaluation of the impact of temperature on hard disk drive failures," *ACM Transactions on Storage (TOS)*, 2013.
[29] K. Ebrahimi, G. F. Jones, and A. S. Fleischer, "A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities," *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 622–638, 2014.
[30] A. Sridhar, A. Vincenzi, D. Atienza, and T. Brunschwiler, "3d-ice: A compact thermal model for early-stage design of liquid-cooled ics," *IEEE Transactions on Computers*, vol. 63, no. 10, 2014.
[31] M. Zapater, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun, "Leakage and temperature aware server control for

improving energy efficiency in data centers," in *Proc. Conference on Design, Automation and Test in Europe (DATE)*. IEEE, 2013.
[32] 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.
[33] S. Sankar, M. Shaw, and K. Vaid, "Impact of temperature on hard disk drive reliability in large datacenters," in *IEEE/IFIP 41st International Conference on Dependable Systems & Networks*. IEEE, 2011.
[34] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *ACM SIGARCH Computer Architecture News*, vol. 34, no. 2. IEEE, 2006.
[35] A. Pahlevan, J. Picorel, A. P. Zarandi, D. Rossi, M. Zapater, A. Bartolini, P. G. Del Valle, D. Atienza, L. Benini, and B. Falsafi, "Towards near-threshold server processors," in *Proc. Conference on Design, Automation & Test in Europe (DATE)*. IEEE, 2016.
[36] Q.-G. Wang, T.-H. Lee, H.-W. Fung, Q. Bi, and Y. Zhang, "PID tuning for improved performance," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 4, pp. 457–465, 1999.
[37] J. Kim, M. M. Sabry, D. Atienza, K. Vaidyanathan, and K. Gross, "Global fan speed control considering non-ideal temperature measurements in enterprise servers," in *Proc. Conference on Design, Automation & Test in Europe (DATE)*. IEEE, 2014.
[38] C. E. Bash, C. D. Patel, and R. K. Sharma, "Dynamic thermal management of air cooled data centers," in *The 10th Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronics Systems (ITherm)*. IEEE, 2006.
[39] L. Parolini, B. Sinopoli, B. H. Krogh, and Z. Wang, "A cyber–physical systems approach to data center modeling and control for energy efficiency," *Proc. the IEEE*, vol. 100, 2012.
[40] J. Meza, M. A. Shah, P. Ranganathan, M. Fitzner, and J. Veazey, "Tracking the power in an enterprise decision support system," in *Proc. ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED)*. ACM, 2009, pp. 261–266.
[41] C. S. Chan, B. Pan, K. Gross, K. Vaidyanathan, and T. Š. Rosing, "Correcting vibration-induced performance degradation in enterprise servers," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 3, pp. 83–88, 2014.
[42] K. Skadron, T. Abdelzaher, and M. R. Stan, "Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management," in *Proc. International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2002.
[43] A. Coskun, J. Ayala, D. Atienza, T. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," in *Proc. Conference on Design, Automation & Test in Europe (DATE)*. IEEE, 2009.
[44] R. Ayoub, R. Nath, and T. Rosing, "JETC: Joint energy thermal and cooling management for memory and CPU subsystems in servers," in *IEEE 18th International Symposium on High Performance Computer Architecture (HPCA), 2012*. IEEE, 2012, pp. 1–12.
[45] R. Ayoub, K. Indukuri, and T. S. Rosing, "Temperature aware dynamic workload scheduling in multisocket CPU servers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1359–1372, 2011.
[46] C. Hankendi, A. K. Coskun, and H. Hoffmann, "Adapt&cap: Coordinating system-and application-level adaptation for power-constrained systems," *IEEE Design & Test*, vol. 33, no. 1, 2016.
[47] "K-Series Electrodynamic Shakers | Unholtz Dickie." [Online]. Available: http://www.udco.com/products/electrodynamic-shaker-systems/k-series/
[48] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," *Advances in neural information processing systems*, vol. 15, 2003.
[49] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, Dec. 1953.
[50] D. Economou, S. Rivoire, C. Kozyrakis, and P. Ranganathan, "Full-system power analysis and modeling for server environments," in *International Symposium on Computer Architecture*. IEEE, 2006.
[51] D. Tsirogiannis, S. Harizopoulos, and M. A. Shah, "Analyzing the energy efficiency of a database server," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*. ACM, 2010, pp. 231–242.
[52] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Microarchitecture, 2009. 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42)*. IEEE, 2009.
[53] Y. Liu, R. P. Dick, L. Shang, and H. Yang, "Accurate Temperature-dependent Integrated Circuit Leakage Power Estimation is Easy," in *Proc. Conference on Design, Automation & Test in Europe (DATE)*. IEEE, 2007.
[54] J. M. Borwein and A. S. Lewis, *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.
[55] SPEC, "SPEC CPU 2006 Benchmarks," 2006. [Online]. Available: https://www.spec.org/cpu2006/
[56] *IPMI Intelligent Platform Management Interface Specification*, 2nd ed., Dell, Intel, Hewlett-Packard, NEC, 2009.
[57] D. Valerio and J. S. da Costa, "Tuning of fractional PID controllers with Ziegler-Nichols-type rules," *Signal Processing*, vol. 86, no. 10, 2006.



Christine S Chan received her PhD degree from the Electrical and Computer Engineering Department at the University of California, San Diego (UCSD) in 2017. She is currently with Qualcomm Technologies, Inc. She received her BS from the University of Illinois, Urbana-Champaign in 2011, and MS from UC San Diego in 2013, both in Computer Engineering. Her main research interest is in optimizing energy efficient embedded systems according to both human and machine context.



Alper Sinan Akyürek received his PhD degree from the Electrical and Computer Engineering at UCSD in 2016. His current work is on control and optimization of energy efficiency in the smart grid. He obtained his M.Sc. and B.Sc. degrees in Electrical and Electronics Engineering from Middle East Technical University in 2011 and 2008, respectively. Prior to PhD, he worked as a Senior Design Engineer on Wireless Networks at Aselsan, Turkey.

Baris Aksanli is an assistant professor at Electrical and Computer Engineering department of San Diego State University. He was a postdoctoral researcher in the Computer Science and Engineering Department at UCSD. He received his PhD and MS degrees in Computer Science from UCSD, and two BS degrees in Computer Engineering and Mathematics from Bogazici University, Turkey. His research interests include energy efficient cyber physical systems, human behavior modeling for the Internet of Things, big data for energy efficient large-scale systems. He won the Internet2 IDEA Award with his work in Lawrence Berkeley National Laboratory and Spontaneous Recognition Award from Intel.

Tajana Šimunić Rosing is a Professor, a holder of the Fratamico Endowed Chair, and director of System Energy Efficiency Lab at UCSD. She currently heads the SmartCities effort as a part of DARPA and industry funded TerraSwarm center. During 2009-2012 she led the energy efficient datacenters theme as part of the MuSyC center. Her research interests are energy efficient computing, embedded and distributed systems. Previously, she was a full time researcher at HP Labs and part time researcher at at Stanford University. In 2001, she finished her PhD at Stanford University, concurrent with a Masters in Engineering Management. Her PhD topic was Dynamic Management of Power Consumption.