Thermal Aware Server Provisioning And Workload Distribution For Internet Data Centers*

Zahra Abbasi, Georgios Varsamopoulos and Sandeep K. S. Gupta
Impact Laboratory
School of Computing, Informatics and Decision Systems Engineering
Arizona State University
http://impact.asu.edu/

ABSTRACT

With the increasing popularity of Internet-based information retrieval and cloud computing, saving energy in Internet data centers (a.k.a. hosting centers, server farms) is of increasing importance. Current research approaches are based on dynamically adjusting the *active server set* in order to turn off a portion of the servers and save energy without compromising the quality of service; the workload is then distributed, conventionally equally (i.e. balanced), across the active servers. Although there is ample work that demonstrates energy savings through dynamic server provisioning, there is little work on thermal-aware server provisioning. This paper provides a formulation of the thermal aware active server set provisioning (TASP), in a nonlinear minimax binary integer programming form, and a series of heuristic approaches to solving them, namely MiniMax, bb-sLRH, CP-sLRH and sLRH. Furthermore, it introduces thermal-aware workload distribution (TAWD) among the active servers. The proposed heuristics are evaluated using a thermal model of the ASU HPCI data center, while the request traffic is based on real web traces of the 1998 FIFA World Cup as well as the SPECweb2009 suite. The TASP heuristics are found to outperform a power-aware–only server set selection scheme (CPSP), by up to 9.3% for the simulated scenario. The order of achieved energy efficiency is: MiniMax (9.3% savings), CP-sLRH (9.2%), bb-sLRH (8.6%), sLRH (5.8%), compared to CPSP.

Categories and Subject Descriptors

D.4.7 [**Operating Systems**]: Organization and Design—*Distributed systems, Hierarchical design, Interactive systems*; D.4.8 [**Operating Systems**]: Performance; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems, Performance evaluation*

General Terms

Design, Management, Performance

Keywords

Data center energy saving, thermal aware server provisioning, thermal aware workload distribution

*This research has been funded in parts by NSF CNS grants #0834797 and #0855277, and by Intel Corp.

1. INTRODUCTION

With the increasing prevalence[1] of Internet-based computing services (partly due to the social networking service boom), there is a consequent increase in the number and size of server farms. Thus, the energy consumption of those server sites will keep growing, and so will the importance of saving energy in them [1–5].

Recent studies [1–3] have demonstrated that energy savings in data centers can be achieved through workload or equipment management techniques using software. The most prominent methodologies proposed in this area are: (i) ***dynamic active server*** provisioning by turning off unnecessary servers [1,4,6], (ii) ***dynamic voltage frequency scaling (DVFS)*** of the servers [7, 8], and (iii) ***thermal-aware workload placement*** [2, 3].

The effectiveness of the first two methods is based on a) the traffic intensity difference between low periods and *workload peaks* which is normally about two to three times as intense [1, 8, 9]; and b) current computing systems consuming significant amount of power at idle compared to being turned off or throttled down [1]. The effectiveness of the third method is based on the non-uniformity of the heat dissipated and recirculated in an air-cooled data center room; therefore, selecting the servers that have the least thermal impact saves energy [2, 3].

The ***state-of-the-art workload scheduling practice*** in Internet data centers is to employ a dynamic active server set scheme at a coarse time granularity (about one hour) with a dispatcher that aims for performance-oriented load balancing across the active set. The *quality of service (QoS)*, defined on the *service-level agreement (SLA) violations*, is maintained by imposing per-server *upper limits* (a.k.a. *caps*) on the CPU utilization level and on the workload that can be dispatched to each server (as shown in §3.4.1, there is a quantitative mapping between utilization and workload). The number of active servers is usually overestimated to prevent service degradation. Related research work shows that dynamic server provisioning saves energy [1, 4, 6, 10], however analytical formalizations focus mainly on estimating the peak traffic [1, 10] leaving the server selection problem as a "homework". The most up-to-date server selection proposes *power-aware* server selection schemes [1, 9], with considerable savings over power-oblivious selection schemes (e.g. random or based on computing performance).

It has been shown in the high performance computing (HPC) do-

[1] `http://www.internetworldstats.com/stats.htm`

main that plain power awareness cannot account for thermal phenomena in the data center room, such as *heat recirculation*, which have a considerable effect on the energy efficiency of a data center. *Thermal awareness*, though, takes into account these phenomena, and there have been studies that enhance HPC scheduling with thermal awareness [2, 3, 11]. Typically, HPC and Internet data centers have similar physical layouts, therefore presumably they exhibit the same thermal phenomena.

The ***approach in this paper*** aims to improve the aforementioned active server set selection with thermal awareness in the analytical formulation of the problem, and with thermal-aware ranking metrics of servers (based on the least recirculated heat metric [2]) so that heuristics can select the servers into the active set.

1.1 Overview of results and contributions

The first main ***contribution of this paper*** is the introduction of thermal awareness into the active server set selection problem (i.e. TASP) through an analytical formulation. This formulation is expressed as a non-linear minimax binary integer programming problem (Eq. 8), where the vector of binary unknowns represents which servers are to be selected (true) or not (false). The integer programming nature of the problem derives from the heat recirculation model which is expressed as a matrix of heat contribution coefficients among servers. The non-linearity comes from the respective non-linearity of the energy efficiency of the cooling equipment, where as the minimax nature comes from the non-uniform effect of heat recirculation among the servers.

The second main contribution is the introduction and evaluation of four heuristics to solve TASP: (i) **MiniMax**, which runs a sequential quadratic programming technique on the continuous-domain version of the problem and then "binarizes" the vector; (ii) **bb-sLRH**, which runs a branch-and-bound technique on a simplified variant of the binary integer formulation, using a least recirculated heat ranking metric (sLRH); (iii) **sLRH**, which runs a one-time rank-and-sort technique on the servers using the sLRH metric; and (iv) **CP-sLRH**, which orders the equipment according to their computing power efficiency and then applies sLRH ranking in each group of servers with the same efficiency.

For the evaluation process, we used a recirculation matrix model based on the ASU HPCI data center, which is a two-row, cold-aisle/hot-aisle air-cooled data center, and recirculates up to 34% of the heat produced by its 200KW computing equipment. The selected request traffic is based on the web traces from the 1998 FIFA World Cup website [12]. As comparison reference, a generic *computing power server provisioning* scheme (CPSP) is used in the simulations. MiniMax is found to provide the best energy savings (9.3%) (albeit with the worst running time), followed by CP-sLRH (9.2%), bb-sLRH (8.6%) and sLRH (5.8%), all with respect to CPSP. Additionally, the simulation section provides a study of the combined performance of TASP with *thermal-aware workload distribution* (TAWD), which offers additional 3% cooling savings that translate into an extra 1% of savings for the given traffic trace.

1.2 Organization of this paper

The rest of the paper is organized as follows: §2 reviews and organizes the related work into a two-tier management architecture, and motivates the need for thermal-awareness. §3 describes the layout of a data center, describes the reference two-tier scheme with load balancing, gives an overview of the thermodynamic modeling of a data center, and sets the foundation toward the integer programming nature of the problem definition. §4 describes the thermal-aware active server set selection (TASP) and thermal-aware workload distribution (TAWD) problems. §5 presents the various heuristics to solve TASP (i.e., MiniMax, bb-sLRH, CP-sLRH and sLRH), as well as a brief description of the solution to TAWD. §6 presents the simulation-based evaluation of the TASP heuristics, with respect to CPSP and no server provisioning, and also their performance when TAWD is enabled. We conclude in §7.

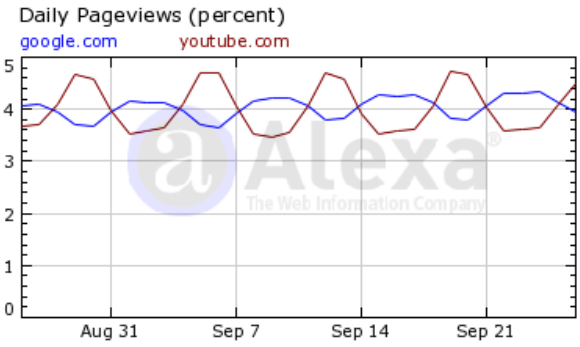

Figure 1: Demonstration of the variation and cyclic behavior of web traffic for two popular web sites (source: www.alexa.com)

2. RELATED WORK AND MOTIVATION

2.1 The overarching problem: saving energy

There exist various energy-saving techniques for data centers, that range from low, hardware level low-power electronics to high, facilities-level energy-efficient design [13] and recapturing the waste heat [14]. A medium-sized data center of 1000 ft^2 can produce the heat at a rate of 3 MW; cooling may require an additional power as high as the heat produced, thus potentially doubling the end-consumption [15]. Therefore, techniques that reduce the cooling needs can yield considerable savings.

There is related work on software-based management to save energy, roughly classified into power-aware and thermal-aware techniques of power, workload and cooling management. This paper belongs to the latter group of studies, and focuses on the problem of saving energy for Internet data centers through thermal aware scheduling. This section gives an overview of the traffic characteristics, how previous work exploited the dynamic, bursty nature of the traffic to save energy, and a recent work on thermal-aware scheduling for HPC batch-job data centers.

2.2 Opportunity: web traffic variation

Intensity variation in web traffic has been witnessed by several research efforts [5, 9, 16]. The variation originates from the size variability of files communicated which forms a fine-scale variation (fluctuation in time scale of a few seconds), and user behavior which forms a coarse-scale (daily or weekly) cyclic variation (see example in Fig. 1). Bradford and Crovella [16] analytically model the behavior of one web user and then use an I.I.D. collection of that model to assess the impact on a web server. Their model is based on a heavy-tailed distribution of file sizes on the Internet and a heavy-tailed distribution of users' thinking time. Chen *et al.* [1] profiled the pattern of Windows Live Messenger load and observed a cyclic behavior.

2.3 Related work on saving energy

2.3.1 Dynamic resizing of the active server set

The concept of active server set is based on the fact that computing resources (i.e. servers) in data centers are usually provisioned for the peak workload. Very few workload peaks match the data center's capacity; a significant portion of the systems would be under-utilized for most of the time if all servers were active all the time. Chen *et al.* [1] proposed *dynamic server provisioning* (i.e. a dynamic active server set scheme) for long-lived TCP-based services. Their study used data traces of Windows Live Messenger, and built a forecasting model to periodically (around 30 minutes) estimate the number of required servers, and save energy based on turning off the unnecessary servers. They balance the load among active servers in slots (around 5 seconds), through balancing the per-server live connections.

Dynamic resizing of the active server set in web hosting centers is also proposed by Chase *et al.* [4], where the center provides different level of services for different customers. They use an economic approach, where services "bid" for resources based on their SLA utility function with the objective of maximizing the profit according to a cost-benefit utility. Load is equally distributed among the servers using round-robin dispatching. Kusic *et al.* [10] propose a dynamic virtual server provisioning scheme by formulating hierarchical optimization problems consisting of both active server set resizing risks and service degradation. The problems are solved dynamically using look-ahead control.

The main design challenge of this group of work is the *period* (i.e. granularity) of the decision making. A low-bounding factor on this period is the delay of server state transition between off and on, typically considered to be a couple of minutes. An upper-bounding factor is the inefficiency in active server set selection caused by the coarser granularity in decision making, leading to energy wastage.

2.3.2 High level DVFS schemes

DVFS, which is available in modern general-purpose CPUs (e.g. as Intel's *SpeedStep* and AMD's *Cool'n'Quiet* and *PowerNow!* technologies) is another studied solution. It is an adaptation mechanism, which adjusts the power scale of a CPU according to the workload in a computing system (the ACPI standard defines those power states as the performance "P" states).

Several papers investigated the control of DVFS by a high-level, global-view software to save energy for data centers [5,7,8]. A control based DVFS policy combined with request batching proposed in [7], which trades off system responsiveness to power saving and adopts a feedback control framework to get a specified response time level. Ranganathan *et al.* [8] proposed a DVFS control to decrease the energy consumption of an enterprise data center based on the estimation of peak utilization of a web server. They argue, the utilization of a web server, serving multiple web applications, should be adapted by the sum of peak traffic of all applications instead of the peak traffic of an individual application.

2.3.3 Hybrids of active server set and DVFS

A hybrid method of resizing active server set and DVFS is proposed by Chen *et al.* [5]. They argued that overestimation of the number of required servers should be compensated by using DVFS in smaller time slots. They developed some methodologies to control CPU power consumption using DVFS.

2.4 New opportunity: thermal aware workload placement

Thermal-aware scheduling has been proposed in some works [2, 3, 11]. Moore *et al.* [3, 17], and Bash and Forman [18] show that thermal aware workload placement can save energy. Mukherjee and Tang *et al.* [2, 11] model the heat that, inefficiently, is recirculated among the servers; using this model, they propose spatio-temporal thermal-aware job scheduling algorithms for HPC batch job data centers. One of their proposed *spatial scheduling* (i.e. *job placement* or *server assignment*) technique is the least recirculated heat (LRH) ranking method, which ranks and sorts the servers according to how much of their produced heat is recirculated, and assigns (or "places") the jobs to the low-ranking servers. While the above research work shows the performance of thermal-aware workload placement in an HPC batch job environment, thermal-aware workload distribution for Internet data centers has not been studied yet.

2.5 Conclusions from the literature review

The related work shows that software-level energy savings in data centers lies in the scheduling of the incoming workload. The state-of-the-art architecture for Internet data centers is a two-tier architecture with dynamic active server provisioning at tier-one and an equal load balancing scheme at dispatch level (tier-two), with OS-driven DVFS optionally enabled at each node. This architecture will form the basis of our comparison in the simulation section and defines three decisions that affect the performance and energy: (i) how many active servers are required in each time interval, (ii) which servers should be chosen in the active set, and (iii) how workload should be distributed among the active servers, to save energy and maintain the performance.

The related work also suggests that the active server set selection is based on the estimation of the peak workload, which in is purposefully overestimated to reduce the probability of SLA violations.

3. SYSTEM MODEL

We assume a virtualized heterogeneous data center, where all servers are capable of running any web application albeit at a different speed and power consumption. This section models the data center physical layout, the operational architecture, energy consumption as well as the performance. Also, we assume an incoming workload that consists of short requests (or transactions). These transactions usually take up to few hundred milliseconds to complete (e.g. HTTP requests). Furthermore, the arrival rate may reach a few thousands requests per second.

3.1 Physical layout of a data center

In contemporary data centers, computing servers are organized in rows of racks of blade systems organized in chassis. The equipment is arranged so that, in each aisle between two rows, it is either front panels or back panels are facing each other; this is called the hot aisle / cold aisle arrangement. In this paper, we refer to a chassis as a computing node. Computing servers consume power, according to their hardware characteristics and computational workload, thus heating the air in the room. Cooling is provided by the Computing Room Air Conditioning (CRAC), which pushes cold air through perforated tiles in the raised floor [3].

3.2 Abstract two-tier architecture

As pointed out in the related work section, (§2) we assume a two-tier, global-view, centralized control software architecture as shown

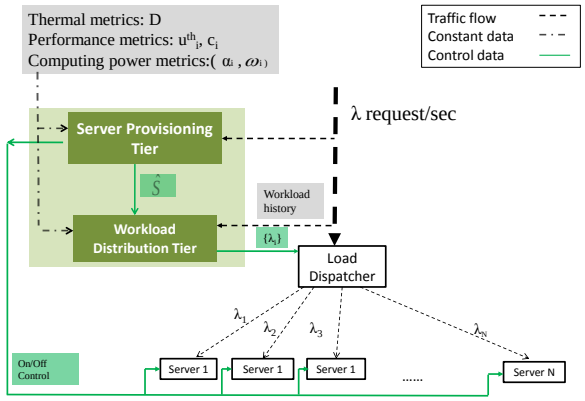

Figure 2: Two-tier architecture for thermal aware management of Internet data centers (server provisioning and workload placement).

in Fig. 2. Tier 1 (T1), the server provisioning tier, iteratively decides the active server set in coarse-time intervals called *epochs*. Let $S=\{s_i\}$, $i=1\ldots N$ be the server set. The T1 controller at the beginning of each epoch estimates $n \leqslant N$ active machines and choose the active server set $\hat{S}$, where $\hat{S} \subseteq S$, $\hat{S} = \{s_i,\ 1 \leqslant i \leqslant N\}$, and $\|\hat{S}\| = n$. Due to the overheads of removing the servers from the active set, e.g. power control and releasing reserved computing resources, an epoch is assumed to be around half an hour.

The controller of Tier 2 (T2), the workload distribution tier, operates at fine-time intervals called *slots* (around 1-10 seconds) and decides on the distribution (i.e. partitioning) of the workload among the active servers. If the average request arrival rate at the m^{th} slot is λ_m, the T2 controller determines β_i and λ_{im}, for all $s_i \in \hat{S}$ where $\sum_{s_i \in \hat{S}} \lambda_{im} = \lambda_m$ and $\lambda_{im} = \beta_i \lambda_m$ such that the SLA performance requirement (usually expressed as a percentile guarantee on the response time) is met and energy consumption is minimized.

3.3 Data center energy consumption model

This subsection provides an overview of the thermodynamic modeling of data centers in [2], adapted to the specifics of the aforementioned two-tier architecture. The total energy consumption of a data center is the sum of the cooling energy and computing energy [2]:

$$E^{total} = E^{comp} + E^{AC}. \tag{1}$$

Cooling energy of the CRAC is modeled by its *coefficient of performance* (CoP), which is the ratio of the heat removed over the work required to remove that heat. A higher CoP means more efficient cooling, and usually the higher the required operating temperatures the better the CoP. The highest CRAC output temperature is limited by the servers' *redline temperature*, as specified by their manufacturer. Heat distribution in a data center room is modeled as a matrix $\boldsymbol{D}=\{d_{ij}\}_{N\times N}$ of coefficients. Each element d_{ij} of this matrix is the coefficient of heat that is distributed from server i to server j [11] (this matrix also converts heat to temperature). Let P_m^{comp} be the total computing power at slot m, and T^{red} the equipments' red line temperature; then the cooling energy at the m^{th} slot is modeled as [2]:

$$E_m^{AC} = \frac{P_m^{comp}}{CoP(T^{red} - \max_i\{DP_m^{comp}\})}t, \tag{2}$$

where, the function max makes sure that the supplied temperature of CRAC does not exceed from the affordable temperature of the hottest equipment. The next step is to model the computing power.

Computing power can be calculated through CPU utilization which is an indication of total power consumption of a typical server [19]. The total power consumption of active servers ($\forall s_i \in \hat{S}$), at a slot m, having CPU utilization of u_{im} can be written as:

$$P_m^{comp} = \sum\nolimits_{s_i \in \hat{S}} (\omega_i + \alpha_i u_{im}), \tag{3}$$

where ω_i denotes power consumption of server i at idle state, and α_i represents extra power consumption at full utilization for each server i. Respectively, $\boldsymbol{w}$ and $\boldsymbol{a}$ denotes vector form of these scalar computing power parameters. This linear computing power consumption model has been derived from experimental measurements done on blade server systems, published in the paper [19]. The error of the linear projection from the actual recordings was about 3% (i.e. about 30W for a 1KW system).

Applying Eqs. 2 and 3 to Eq. 1, the total energy consumption at the m^{th} slot becomes:

$$E_m^{total} = \left(1 + \frac{1}{\text{CoP}\left(T^{\text{red}} - \max\limits_{s_i \in \hat{S}}\{\boldsymbol{D}(\boldsymbol{w} + \boldsymbol{a} \odot \boldsymbol{u}_m)\}\right)}\right) \sum_{s_i \in \hat{S}} (\omega_i + \alpha_i u_{im})t, \tag{4}$$

where $\boldsymbol{u_m}$ represents vector form of utilization of servers. Also the operator $\odot$ is defined such that $\boldsymbol{a} \odot \boldsymbol{u_m}$ is a vector $\langle \alpha_i u_{im} \rangle$, $s_i \in \hat{S}$. Note that, *since the utilization vector u_m directly depends on the workload distribution to the servers, the energy consumption at the m^{th} slot also depends on the workload distribution*.

3.4 Performance modeling

In Internet data centers, performance is usually expressed in throughput, response time and turn-around time. In this context, the SLA statistically bounds the response time:

Prob[*response_time* > response_threshold$_{\text{SLA}}$] < probability_threshold$_{\text{SLA}}$.

3.4.1 SLA, performance and utilization

Although web traffic is not CPU-intensive, related research has identified that the CPU utilization level is strongly correlated to the QoS; specifically, the SLA is violated beyond a CPU utilization point [4].

The aforementioned correlation is observed in the following experiment as well. We configured one computer as web server and another computer as the client generating TCP-based requests on files with size distribution ranging from 0.3KB to 90KB, in accordance to a study on the file size distribution of web image content [20]. Both the web server and client are dual-CPU dual-core E7520-chipset "Sossaman" Xeon LV systems. The average turnaround time and the web servers' CPU utilization over the posed workload (measured as arrival rate) is shown in Fig. 3. It can be seen that, the turnaround time is constant until the utilization reaches to around 20% (or the arrival rate reaches to 2000 requests per second) and then it goes up and even fluctuates. This experiment shows that the quality of service of Internet requests in terms of delay can be guaranteed if a server is not utilized up to a threshold point. The amount of threshold point depends on the hardware capacity of servers and the type of requests. Therefore, we consider that ***by***

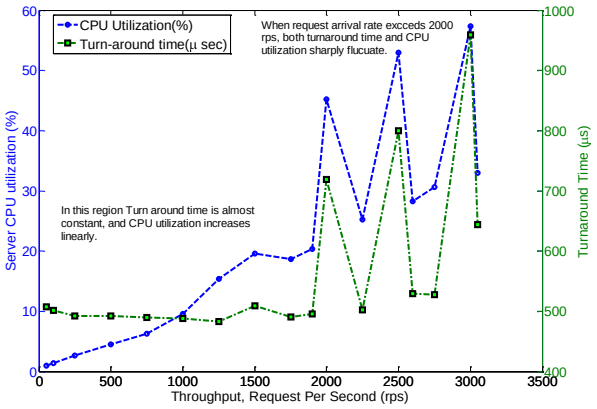

Figure 3: Turnaround time and CPU utilization versus throughput.

posing a bound to the CPU utilization, (i.e. preventing overloading of a server such that its CPU utilization doesn't go beyond a threshold value), we automatically pose a bound to the SLA violation rate. This is an important observation as CPU utilization levels are easier to track than response time.

Thus, performance constraint of a server i at m^{th} slot can be written in the following form:

$$u_{im} = c_i \lambda_{im} \leqslant u_i^{\text{thres}}, \tag{5}$$

where c_i is the average utilization that the *unit request rate* (i.e. 1 req/sec) imposes on a server i, and u_i^{thres} is the CPU threshold for i, which depends on its hardware characteristics and type of requests. Hence the maximum workload arrival rate of a server i can be expressed as:

$$\lambda_i^{max} = \frac{u_i^{\text{thres}}}{c_i}, \tag{6}$$

4. FORMULATION OF THE OPTIMIZATION PROBLEMS

This paper addresses the active server set selection at tier one. Since the utilization of servers affects both the computing and cooling energy, the problem can be formulated as an optimization problem so as to minimize the total energy consumption under the utilization thresholds as follows:

T1 (server provisioning) problem: Given a data center with the server set S with N servers, for an epoch with length Lt, where L is the number of t-length slots in an epoch, how can the active set $\hat{S} \subseteq S$, where $|\hat{S}| = n \leqslant N$, be chosen to minimize the total energy E^{total}?

The size of active server set $\hat{S}$ at the k^{th} epoch depends on the peak arrival rate of that epoch (Λ_k^{peak}):

$$\sum_{s_i \in \hat{S}} \frac{u_i^{\text{thres}}}{c_i} \geqslant \Lambda^{peak}, \tag{7}$$

which means that during an epoch, the threshold utilization of active servers should afford the peak workload arrival rate. Let the binary vector $\boldsymbol{x}$ be the choice of servers as members of active set, hence thermal-aware active server selection can be represented in the following optimization problem of *finding the vector* $\boldsymbol{x}$:

Minimize

$$(1+\frac{1}{\text{CoP}(T^{red} - \max_i(\boldsymbol{x}^T \boldsymbol{D}(\boldsymbol{w}+\boldsymbol{a}\odot\boldsymbol{u}^{\text{th}}))}) \sum_{i=1}^{N} x_i(\omega_i + \alpha_i u_i^{\text{thres}}) Lt. \tag{8}$$

subject to

$$\sum_{i=1}^{N} x_i \frac{u_i^{\text{thres}}}{c_i} \geqslant \Lambda^{peak} \qquad \text{[performance constraint]}$$
$$x_i \in \{0,1\}, \quad \forall i = 1 \ldots N$$

T2 (workload distribution) problem: Given a data center active server set $\hat{S}$ during an epoch with length Lt, how can one determine $\lambda_{im}, \forall s_i \in \hat{S}, \forall m = 1 \ldots L$, where $\sum_{s_i \in \hat{S}} \lambda_{im} = \lambda_m$, such that the utilization threshold constraint (5) is satisfied and energy consumption is minimized? (see Fig. 2).

There is a capacity constraint stating that, in a given slot m, the total workload λ_m should not exceed the maximum affordable workload by all active servers:

$$\lambda_m \leqslant \sum\nolimits_{s_i \in \hat{S}} \frac{u_i^{\text{thres}}}{c_i}.$$

additionally, there is a performance constraint stating that the portion of workload to be assigned to a server should not exceed the server's maximum affordable workload. Let β_{im} be the portion of workload for server i at m^{th}slot, using Eq. 5 the performance constraint at m^{th} slot can be expressed as:

$$c_i \lambda_{im} = c_i \beta_{im} \lambda_m \leqslant u_i^{\text{thres}}, \forall s_i \in \hat{S}.$$

Using the formulations above, TAWD can be represented as the following optimization problem of *finding the workload distribution weights of* $\boldsymbol{b} = \{\beta_{im}\}$:

Minimize E_m^{total} in Eq. 4, subject to:

$$\lambda_m \leqslant \sum_{s_i \in \hat{S}} \frac{u_i^{\text{thres}}}{c_i}, \qquad \text{[capacity constraint]}$$
$$c_i \lambda_{im} = c_i \beta_{im} \lambda_m \leqslant u_i^{\text{thres}}, \forall s_i \in \hat{S}, \qquad \text{[performance constraint]}$$
$$\sum_{s_i \in \hat{S}} \beta_{im} = 1.$$

The next section describes heuristics to solve TASP and TAWD.

5. THERMAL AWARE SERVER PROVISIONING AND WORKLOAD MANAGEMENT

This section presents our solutions to the thermal aware server provisioning and workload distribution. The pseudo-code is given in Algorithm 1. Active server set resizing, for tier one, and workload distribution, for tier two, are controlled by the event-based procedures OnEpochTimeOut and OnSlotTimeOut, respectively. Thermal aware server provisioning (procedure TASP) as well as thermal aware workload distribution (procedure TAWD) are performed based on the heuristic solutions described below.

Dynamic workload estimation is done using two Kalman filters which estimate the average arrival rate of HTTP requests at each epoch (Λ) and slot (λ_m), respectively[2]. However, to determine the size of active server set for every epoch, the peak arrival rate (Λ^{peak}) is required rather than the average arrival rate (Λ). The challenge of estimating the peak arrival rate (Λ^{peak}) is overcome by following

[2]An *a priori* distribution cannot be assumed, due to the dynamic and fluctuating nature of the Internet traffic.

Table 1: Symbols and definitions

Symbols	Definition
N	The number of total computing nodes
n	The number of active computing nodes
t	the length of slots (in second)
L	The number of slots in a given epoch
i	index of nodes
m	index of slots
k	index of epochs
λ_m	the average workload rate at m^{th} slot
Λ_k	the average workload rate at k^{th} epoch
Λ_k^{peak}	the peak workload rate at k^{th} slot
γ	the "overestimation" factor, used to estimate Λ^{peak}
ω_i	The power consumption of a computing node at idle time
α_i	The power consumption of node i while is full utilizing minus ω_i
$\beta_i m$	the percentage of workloads assigned to the node i at m^{th} slot
λ_{im}	the workload arrival rate of node i at m^{th} slot
u_{im}	the average utilization of node i at m^{th} slot
u_i^{thres}	the threshold utilization of server i
c_i	the average utilization that a unit arrival rate imposes on a node i
d_{ij}	The heat dissipated from node i to node j
p_{im}^{comp}	The power consumption of server i at m^{th} slot
S	The set of all servers
$\hat{S}$	The active server set
$\boldsymbol{\lambda}_k$	The vector λ_m at k^{th} epoch
$\boldsymbol{w}$	The vector $\{\omega_i\}_n$
$\boldsymbol{a}$	The vector α_{i_n}
$\boldsymbol{u}_m$	The utilization vector $u_{im\{}n\}$
$\boldsymbol{p}_m$	The computing power vector $p_{im\{}n\}$
$\boldsymbol{b}_m$	The workload portion vector $\beta_{im\{}n\}$
$\boldsymbol{D}$	The heat recirculation matrix $\{d_{ij}\}_{(n\times n)}$
E_m^{total}	Total energy consumption of data center during a slot m
E_m^{AC}	The total cooling energy during m^{th} slot
P_m^{comp}	Total computing power consumption during m^{th} slot

a method similar to the one in [10]. Let the vector $\boldsymbol{\lambda}_{k-1}$ contain the *observed* average request rates during slots $1 \ldots L$ of the previous epoch, i.e. epoch $k-1$. Then the peak arrival rate is estimated using the standard deviation of the workload, $\sigma(\boldsymbol{\lambda}_{k-1})$, on the previous epoch and the Kalman-estimated Λ_k, as follows:

$$\Lambda_k^{peak} = \Lambda_k + \gamma\, \sigma(\boldsymbol{\lambda}_{k-1})\, \frac{\Lambda_k}{\Lambda_{k-1}}, \tag{9}$$

where γ represents the *scaling* or *overestimation factor*. In the evaluation section, we study TASP and TAWD under various γ values.

5.1 Solutions for tier one problem

To solve the nonlinear optimization problem of TASP, we propose the following approaches.

5.1.1 MiniMax: a minimax solution

The optimization problem of Eq. 8 is a nonlinear minimax optimization problem. Minimax problems can be solved numerically in MATLAB using sequential quadratic programming (SQP), which in turn uses quadratic programming (QP) and quasi-Newton to approximate the Hessian of the Lagrangian function, iteratively. The Hessian is always kept positive so that the problem can be solved in polynomial time. The time complexity of QP is polynomial $O(m^3 M)$, where m is the number of variables and M is the size of input) [21], when the approximated Hessian is positive semidefinite.

The given T1 problem is a discrete minimax[3] problem. For that reason, MiniMax computes a solution in the continuous domain, and then discretizes the vector to the closest discrete solution:

> **Algorithm MiniMax: 1.** Solve the problem in Eq. 8 using a minimax solver such as SQP on the continuous domain; obtain vector $\boldsymbol{v}$. **2.** Sort $\boldsymbol{v}$ in descending order, then chose enough of the corresponding highest-value servers (each element in $\boldsymbol{v}$ corresponds to a server) as the active server set to satisfy the capacity constraint of Eq. 8.

The high complexity of MiniMax limits its use in the online selection of active servers. However, due to it providing a good approximation, it is used for comparison in the evaluation section.

5.1.2 The Least Recirculated Heat (LRH) heuristics

The insight behind the LRH heuristics is based on the observation that the energy is consumed either by the computing equipment or by the cooling equipment (Eq. 4). Therefore, a heuristic would be to minimize sum of one or both of these parameters instead of minimizing the entire energy formulation in Eq. 4.

Minimizing the computing power means minimizing Eq. 3. This can be done by ranking servers according to their computing power and sorting them accordingly (which is the base of the CPSP heuristic). On the other hand, minimizing the heat recirculation is similar to the denominator in Eq. 2, which is :

$$\sum DP_m^{comp} = \sum \left(\boldsymbol{x}^T \boldsymbol{D}(\boldsymbol{w}+\boldsymbol{a}\odot\boldsymbol{u}^{\text{th}}\right). \tag{10}$$

Contrasted to $\max_i DP_m^{comp}$, the summation does not try to minimize the maximum inlet temperature, but instead it tries to minimize the sum of the inlet temperatures. This heuristic way is referred to as Least Recirculated Heat (LRH), first introduced in [2] for HPC batch job data centers. In LRH, each server is ranked on the amount of its output heat that is recirculated and on how loaded the receptor nodes are, recirculation-wise, i.e. a server i rank is defined as:

$$r_i^{\text{LRH}} = \sum_j v_j d_{ij} p_i^{comp}, \text{ where } v_j = \sum_i d_{ij} p_i^{comp}. \tag{11}$$

In this paper, we redefine the ranking to take into consideration the node's computational capabilities; in the HPC-oriented work in [2], the computational performance was expressed and formulated outside the metric's definition. Thus, we define the *scaled LRH* (sLRH) as:

$$\hat{r}_i^{\text{LRH}} = \frac{r_i^{\text{LRH}}}{(u^{\text{thres}}/c_i)^2} = \frac{\sum_j v_j d_{ij} p_i^{comp}}{(u^{\text{thres}}/c_i)^2} \text{ ,where } v_j = \sum_i d_{ij} p_i^{comp}, \tag{12}$$

where, p_i^{comp} is the power consumption of server i, and it can be statically computed by adjusting p_i^{comp} as $p_i^{comp} = (\omega_i + \alpha_i u_i^{\text{thres}})$.

The following three heuristics are proposed with respect to the LRH ranking metric.

Branch and Bound LRH (bb-sLRH): It is clear that to choose most thermal efficient server based on the sLRH ranking, their computing capacity should also be taken into account. Choosing the active server set through total minimizing heat recirculation ($\hat{r}_i^{\text{LRH}}$) can be formulated and solved as Binary Integer Programming (BIP) as follows:

Minimize: $\sum_{i=1}^N \hat{r}_i^{\text{LRH}} x_i$
Subject to:
$\sum_{i=1}^N x_i \frac{u_i^{\text{thres}}}{c_i} \geqslant \Lambda^{peak}$, [Capacity Constraint]
$x_i \in \{0,1\} \forall i = 1..N$

[3] We use the term *minimax* to refer to the class of problems, and *MiniMax* as the name of the specific algorithm defined in this paper.

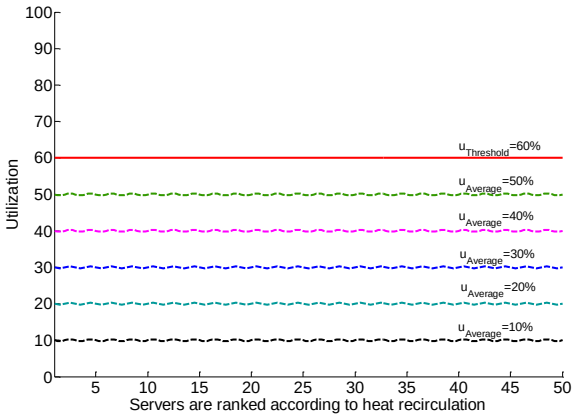

Figure 4: Utilization of servers in load balancing. Server are ranked according to Eq. 11.

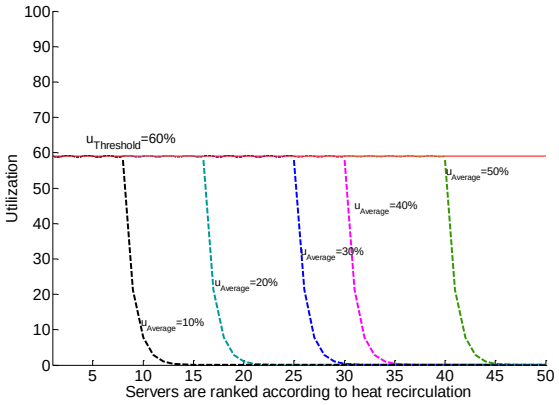

Figure 5: Utilization of servers in TAWD. Servers are ranked according to Eq. 11.

BIP is NP-complete in general, and the branch-and-bound algorithm to solve BIP has exponential computation complexity in the worst case. Therefore, it is used to evaluate other TASP schemes.

Ordered selection by sLRH (sLRH): Another heuristic method to minimize the total heat recirculation is choosing the n lowest ranking servers that satisfy bb-sLRH's capacity constraint above. This is done by starting with the lowest ranking server, i.e. $\hat{S} = \{\arg\min\{\boldsymbol{r}\}\}$, and then iteratively adding servers to $\hat{S}$ until their capacity equals or exceeds Λ^{peak}.

Hybrid of CPSP and sLRH (CP-sLRH): Although both sLRH and bb-sLRH take into account both the heat recirculation and computing power, the quality of their solution in heterogeneous data centers may not be a good approximation when the computing power dominates the cooling power. Hence, we define another heuristic in which servers are first ranked according to their computing efficiency ($p_i^{comp}/u^{\text{th}}/c_i$) and then according to sLRH ($\hat{r}^{LRH}$) within each group of servers of the same computing efficiency. This algorithm is referred to as computing power sLRH hybrid server provisioning (CP-sLRH) in the rest of the paper. Due to its low time complexity, this algorithm is another choice to use in online TASP, as shown in Alg. 1.

5.1.3 CPSP: the baseline algorithm

Most of the previous research takes into account only the computing efficiency in the active server selection. For this reason, we used the power-aware yet thermally oblivious approach of CPSP (computer power server provisioning) as the baseline algorithm to evaluate the efficiency of TASP. In CPSP, servers are ranked based on their computing power efficiency ($p_i^{comp}/u_i^{\text{thres}}/c_i$) only, with disregard to their thermal impact; then a method similar to sLRH above is used to populate the active server set $\hat{S}$.

5.2 Tier two: Thermal Aware Workload Distribution

Energy efficient workload distribution, as described in the T2 problem (§4), compensates energy wasting resulted from inevitable overestimation of the number of active servers (Λ^{peak}). However, it is not time-efficient to solve yet another non-linear optimization problem (this time to distribute workload), especially in slot intervals. Therefore, the sLRH ranking is used to skew workload to the most thermally efficient servers. This method can be clarified by comparing Figs. 4 and 5, which show how workload is distributed using equal *load balancing* (LB), in Fig. 4, and using TAWD, in Fig. 5. In load balancing, the workload is distributed equivalently among servers such that their utilization levels are *equalized* (i.e. *balanced*).

In TAWD, servers with low sLRH will be more likely to be utilized close to the threshold point u^{th} while servers with higher sLRH will be less likely be utilized. In other words, if we have 50 homogeneous servers with 30% average utilization in load balancing, in TAWD, ($\frac{50\times30\%}{60\%}$) servers with least heat recirculation get 60% utilization and others remain idle or will be utilized very low. For ranking, the CP-sLRH method can be used as an alternative to pure sLRH.

Considering the procedures mentioned in Alg. 1, at the beginning of every fine time slot (OnSlotTimeOut procedure in Alg. 1), the upcoming workload is estimated using Kalman filtering technique. Then, in order to determine the scheduling policy the procedure TAWD (see Alg. 1) is called which starts from the most energy efficient server (obtained by sLRH or CP-sLRH ranking), and assigns as big portion of workload (λ_{im}) as it can be afforded by that server according to its performance constraints. It continues to assign workload portions to other servers based on the chosen ranking, in descending order. Upon the arrival of each HTTP request (procedure UponJobArrival in Alg. 1), the request is stochastically assigned according to the skewed $\boldsymbol{\lambda}_m$ vector, thus the workload is similarly skewed toward the low ranking servers.

However, TAWD increases the chance of SLA violation with respect to LB because it concentrates the workload on specific servers, which thusly are more likely to cross the performance threshold u^{th}. This is a direct consequence of TAWD's workload-skewing nature and the possible underestimation of the arrival rate λ_m. Therefore, the dispatching algorithm is configured so that, as long as the observed arrival rate is less than the estimated arrival rate (meaning that TAWD does not overload thermally efficient servers), the dispatcher can dispatch jobs according to the TAWD policy, otherwise it dispatches jobs according to LB (UponJobArrival in Alg. 1).

Algorithm 1 Thermal aware server provisioning and workload distribution

procedure INITIALIZATION( )
  Rank servers based on TASP algorithm (sLRH, CP-sLRH)
  Set up epoch controller timer:*OnEpochTimeOut* (i.e. Lt)
  Set up slot controller timer:*OnSlotTimeOut* (i.e. t)
end procedure

procedure ONEPOCHTIMEOUT( )
  Estimate average arrival rate (Λ) and calculate Λ^{peak} using (Eq. 9)
  $\hat{S}$ = call TASP(Λ^{peak})
  Switch On/Off servers if required
end procedure

procedure TASP(Λ^{peak})
  $\hat{S} \leftarrow \{\}$;
  while $\sum_{s_i \in \hat{S}} \frac{u_i^{\text{thres}}}{c_i} \geqslant \Lambda^{peak}$ is not satisfied **do**
    Add next lowest ranking server s_i (using sLRH or CP-sLRH ranking):
    $\hat{S} \leftarrow \hat{S} \cup s_i$
  end while
end procedure

procedure ONSLOTTIMEOUT( )
  Estimate upcoming workload arrival rate (λ_m)
  $\boldsymbol{b}$=Call *TAWD(λ_m)*
end procedure

procedure TAWD(λ_m)
  $\beta_i \leftarrow 0, \forall i = 1 \ldots n$
  while $\sum_{i=1}^{n} \beta_i ! = 1$ **do**
    Choose the next most thermal efficient server ($s_i \in \hat{S}$)
    Maximize β_i in the following equations:
    1:$\beta_i = \frac{\hat{u}_i^{\text{thres}}}{c_i} \frac{1}{\lambda_m}$
    2:$\sum_{i=1}^{n} \beta_i = 1$
  end while
end procedure

procedure UPONJOBARRIVAL($\boldsymbol{b}$)
  if Estimated $\lambda_m \leqslant$ observed λ_m **then**
    Dispatch job a server according to TAWD policy ($\boldsymbol{b}$)
  else
    Dispatch jobs to a server based on Load Balancing policy
  end if
end procedure

6. EVALUATION

This section presents energy efficiency analysis of the TASP approaches with respect to CPSP under different utilization levels of data center as well as different schemes for estimation of the peak traffic. Additionally, it presents the performance analysis of TAWD with respect to load balancing (LB). For the evaluation purpose we setup a simulation environment in which heat recirculation of a real data center as well as real web traffic are used.

6.1 Simulation Setup

This section describes our simulation environment, i.e. the data center layout, web traffic and server model.

6.1.1 Data Center Profile

Data center profile consists the heat recirculation matrix, and the servers' power and performance models.

The heat recirculation matrix, representing the air flow and heat distribution of a data center room is obtained from a linear model as published in [2, 22] and simulation of the ASU HPCI data center physical layout. The linear model of heat recirculation has been validated using FloVENT CFD simulations, where predicted temperatures from the model are compared with temperatures yielded by the simulation. The average temperature error for the simulated heat flow model is around 0.34 °C in this method. The heat recirculation of the ASU HPCI data center physical layout has been derived by the aforementioned model which is used in our simulation environment.

We simulate the same computing infrastructure that is in the ASU HPCI data center[4]: thirty Dell PowerEdge 1955 chassis (10 blade servers of 4 cores), and twenty Dell PowerEdge 1855 chassis (10 blade servers of dual cores). The maximum possible computing-only power consumption is just above 200KW. The power model of these two computing equipment types, i.e. ω and α, is yielded for I/O intensive jobs, which is appropriate for HTTP-like Internet requests (see Table 2) [2].

[4]This is an equipment "snapshot" reflecting early 2007; the facility has evolved and grown since then.

Table 2: Computing Power Parameters

PowerEdge	ω	α
model 1855	2020	50
model 1955	1590	90

Table 3: CPU threshold violations with respect to γ

γ	Average $\lvert\hat{S}\rvert$	Violation occurrence
1	9	3.34%
3	13	1.37%
6	20	0.03%

6.1.2 Requests Profile and System Model

We analyzed TASP under SPECweb2009 [23] e-commerce web benchmark suite. SPECweb2009 is able to generate e-commerce requests to browse, buy, or sell in the web environment. We set up an Apache web server on a Dual-core Intel Xeon LV system and an additional system as the SPECweb traffic-generating client. While SPECweb is apt at exhibiting small-scale traffic variation, it needs to be dynamically controlled, by a configuration parameter called "SIMULTANEOUS SESSION" to achieve a long-scale traffic variation. In order to adjust this parameter to exhibit large scale traffic variation, we synthesized a part of the 1998 FIFA World Cup web trace [12]. In this model, the large scale traffic intensity of the FIFA's web log (24 hours) is used to determine the distribution of concurrent threads on the SPECweb client. Then, SPECweb is configured to dynamically change the number of concurrent threads according to the given distribution (see Fig. 6). The 24-hour traffic profile derived from the collected logs is scaled up to the number of servers in the simulated data center. Although a day's variation is enough to show the energy-saving benefits of the TASP approach, to make sure that the duration does not affect the validity of our results, we also repeated analysis for the five, highly active consecutive weeks of FIFA's 1998 World Cup web traces (June and the first week in July), traffic which is representative of an active web service (see Fig. 7).

The server utilization is modeled according to the arrival rate and utilization relationship of Eq. 5. The server utilization thresholds, u^{thres}, are set to 60%[5]. Based on the CPU utilization profile col-

[5]This value was determined from anecdotal Web searching. It does not affect the validity of the results but only the amount of savings.

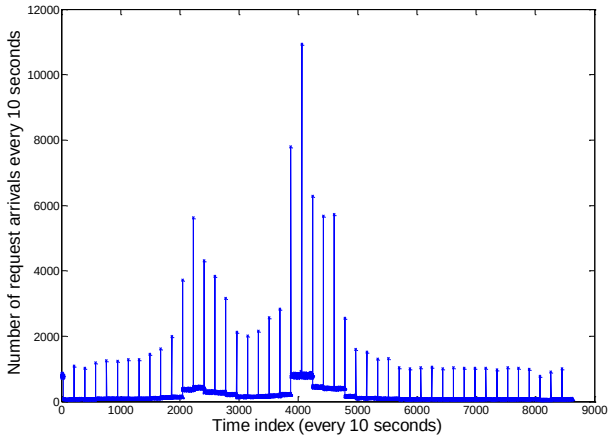

Figure 6: Request arrival rate over time of SPECweb2009 where epoch-level peaks are obtained from the 1998 FIFA World Cup traces (Fig. 7).

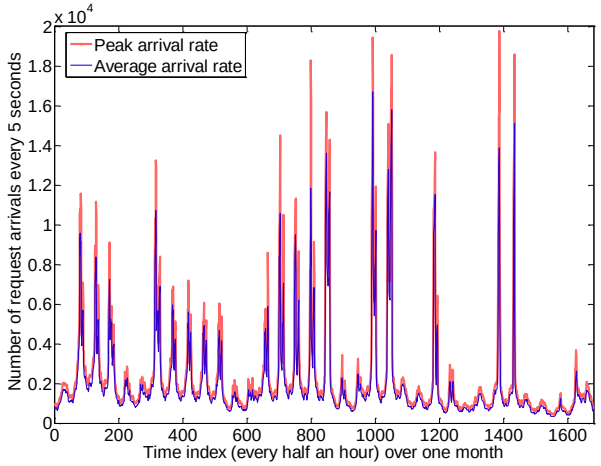

Figure 7: HTTP requests over time, 1998 FIFA World Cup [12].

lected from the experiments above, the c_i parameters (i.e. utilization of a unit arrival request) in the two type of servers are adjusted proportionally to the number of their cores ($c_{i,1855}$ = 0.4, $c_{i,1955}$ = 0.1). Using this setup, the average utilization of our simulated data center under no server provisioning using the above workload is about 8%.

6.1.3 *Dynamic resource provisioning*

Two Kalman filters are trained for five s lots and epochs, respectively. They then begin to estimate the average rates, one at slots (λ_m) and and one epochs (Λ). The number of active servers is estimated using Eq. 9, for γ=1,3,6.

Table. 3 shows the average number of active servers and average CPU utilization violations with respect to γ. The table also indicates the tradeoff between energy savings and quality of service. Note that a smaller active server set causes higher violations but also higher energy saving.

6.2 Thermal aware server provisioning

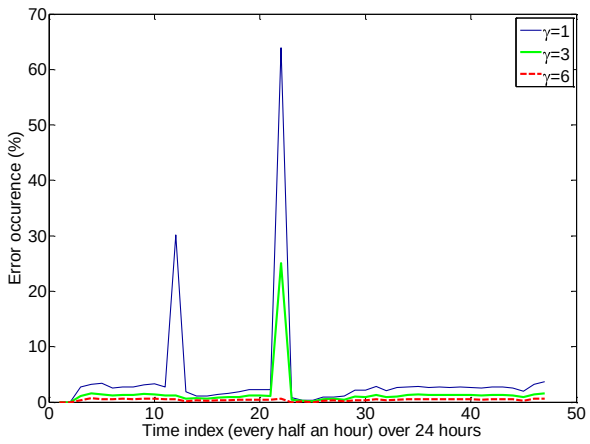

Figure 8: CPU utilization violations with respect to γ over time. Violations for γ=1 are much higher than for the rest values. The total percentage of violations are given in Table 3.

This subsection evaluates the performance of TASP when LB is used. Specifically, it evaluates the energy efficiency of the four approaches (MiniMax, sLRH, bb-sLRH, CP-sLRH) in §5.1, with respect to CPSP. We used the solvers provided by MATLAB to implement and run the MiniMax, (*fminimax*) and branch & bound (*bintprog*) algorithms.

6.2.1 *The performance of various TASP approaches*

Fig. 11 shows that most of the TASP algorithms (except for sLRH) always perform better than CPSP, and can save energy between 8% down to 3.4% with respect to CPSP, depending on the value of γ and the TASP algorithm in question (see Fig.11). Among the TASP approaches, MiniMax which is an approximation of the optimal answer for the energy consumption model (Eq. 4) gives the best energy saving. The quality of the MiniMax solution comes at a high computational cost: the complexity of the QP alone, which is used iteratively in MiniMax, is almost $O(n^4)$. On the other hand, the computation time of the other heuristics (CP-sLRH and sLRH) is linear. As a practical example, CP-sLRH and sLRH take a fraction of second to compute the active server set for a hypothetical data center of 500 chassis, while MiniMax takes around half an hour, all run on a 2.8 GHz Intel Pentium system, as performed in a side experiment. This is an indication of poor scalability of the MiniMax approach. This trade-off between energy savings and execution time is a motivation to look for possible ways to improve the polynomial computation time of MiniMax approach for future work.

The next best algorithm (See Fig. 11), in terms of energy savings, is CP-sLRH, which considers both the computing power efficiency and heat recirculation. Note that CP-sLRH would always give a better solution with respect to CPSP, in any data center; this is because, when power efficiency is equal, it prefers the lower sLRH-ranking, i.e. more cooling-efficient, server.

The bb-sLRH and sLRH tend to save less energy than MiniMax over CPSP. While bb-sLRH always surpasses CPSP in terms of energy saving, sLRH does worse than CPSP for γ = 6. The reason is that the way sLRH is defined may not capture the proper order of ranking in all cases as it would have been computed by an exact so-

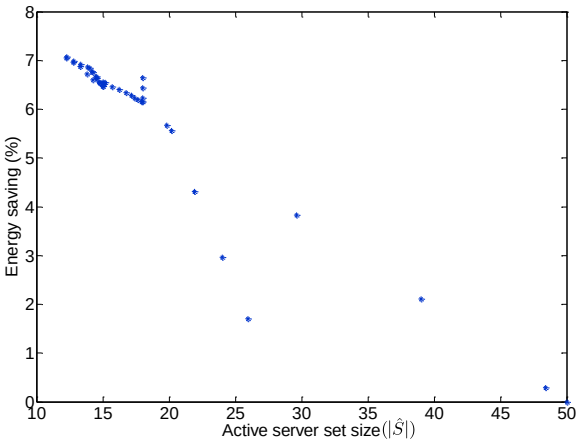

Figure 9: Energy saving of MiniMax over different number of active server size ($|\hat{S}|$) for γ=6. Each mark represents one epoch. Note that energy saving is higher for smaller $|\hat{S}|$.

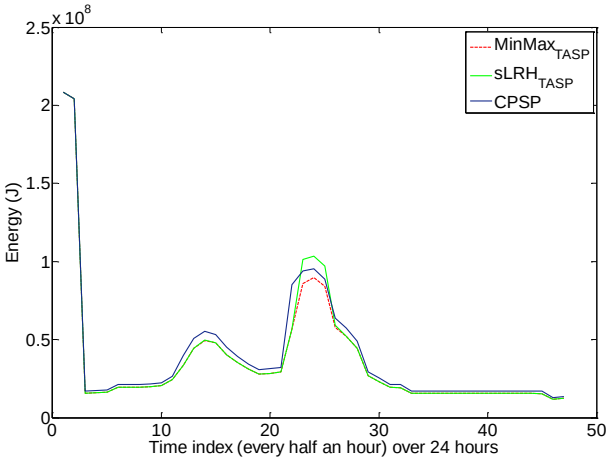

Figure 10: Energy consumption of thermal aware server provisioning scenario over time (intervals in epochs) (γ = 1).

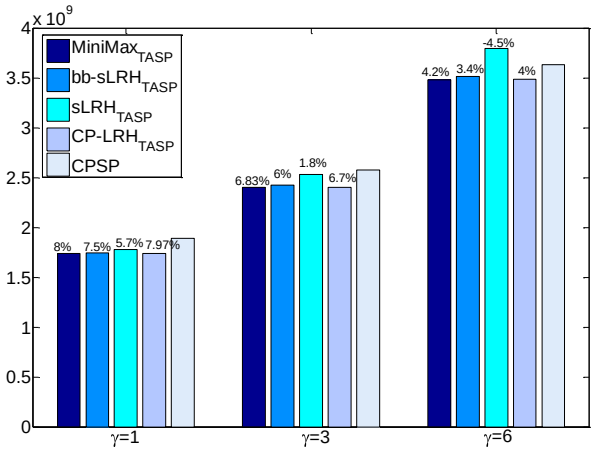

Figure 11: Total energy consumption with respect to server provisioning scenarios. The energy-saving percentages are with respect to CPSP.

lution to the proper problem of Eq 8. In fact, when the contribution of computing efficiency in the total energy consumption surpasses the cooling efficiency, sLRH may perform worse than CPSP. In our results, this happens when $|\hat{S}|$ becomes large. In fact in the small $|\hat{S}|$, thermal efficient servers which are chosen as active server set some computing efficient enough to save energy over CPSP. But this is not true for when $|\hat{S}|$ become large. This is also shown in Fig. 10; the figure shows that, in the peak traffic time, where the $\hat{S}$ becomes large, sLRH performs worse than CPSP. Note that the same figure shows that MiniMax always performs better than CPSP. The bb-sLRH heuristic, which finds the optimal ranking for sLRH for a specific workload, manages to lessen this quality degradation effect: it performs very close to MiniMax in all tested scenarios.

6.2.2 *Energy saving with respect to the overestimation factor γ*

As depicted in Figs. 11 and 12, the energy savings also change with respect to different values of γ . This difference mostly is due to the average active server set size ($|\hat{S}|$). For smaller $|\hat{S}|$, the average efficiency of TASP's active server set is more than what has been chosen by CPSP. This efficiency reaches zero when the active server set equals to the entire server set (See Fig. 9).

Results for the five weeks of FIFA's 98 web trace, shown in Fig. 13, indicate that energy saving has a similar pattern to the 24-hour simulation. The small differences are because of different traffic intensity which leads to different number of active servers. It can be seen that energy saving of MiniMax reaches to 9.3% at $\gamma = 1$, where the average server active set size is around 6 over the one month log. Note that in this result, sLRH always performs better than CPSP. The reason is the smaller active server size ($\hat{S}$) over one month of traffic.

6.2.3 *Discussion on the performance of sLRH and CP-sLRH over different data centers*

The performance of both CPSP and sLRH is affected by the heterogeneity of data centers and equipments. These algorithms yield equal performance for a homogeneous data center. However, depending on the type of equipments and their computing efficiency differences, their performance changes. While CP-sLRH always surpasses CPSP, its performance may be less than sLRH in cases where the computing efficiency of the servers is not as dominant in saving energy as heat recirculation. This happens when the computing efficiency of servers is very similar across the entire set. In contrast, sLRH may even increase the energy consumption with respect to CPSP when the contribution of computing power in the total energy consumption is more than the cooling energy (such as our case).

6.3 Thermal aware workload distribution

The energy savings of TAWD are calculated with respect to a conventional performance-oriented Load Balancing (LB) scheme; CP-sLRH is used to determine the active server set (on which TAWD is applied). The workload-skewing behavior of TASP/TAWD is clearly demonstrated in Fig 14. LB's apparent skewing in the figure is the result of TASP-based active set selection, averaged for all the epochs in one week; in each epoch, LB balances load equally among the active servers.

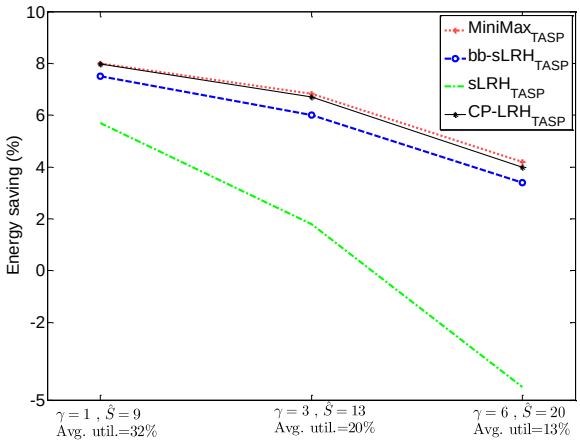

Figure 12: Energy saving with respect to CPSP for different TASP schemes over $|\hat{S}|$. Note that higher utilization yields higher savings.

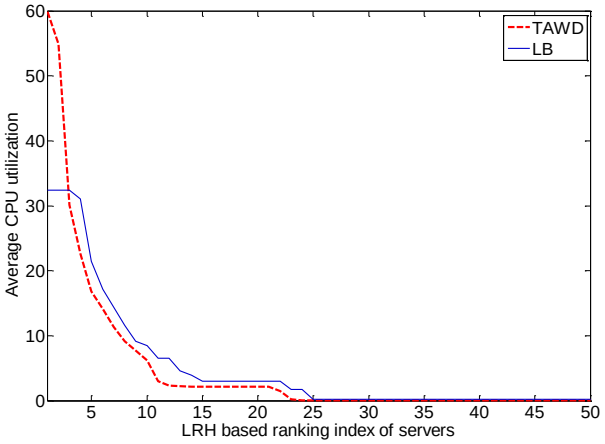

Figure 14: Average data center utilization of each server (over one week), as sorted with respect to LRH. The effects of TAWD's load skewing on the utilization are obvious.

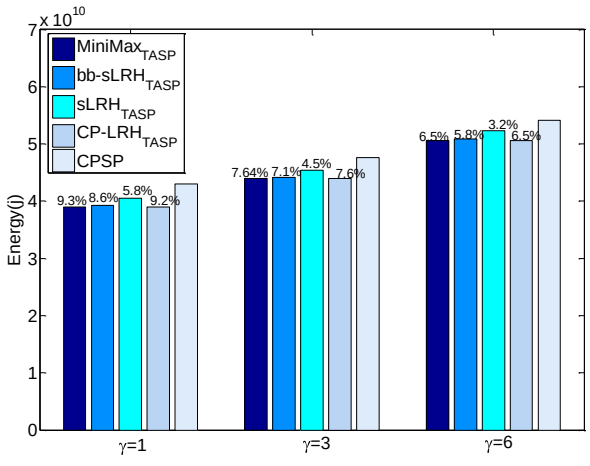

Figure 13: Total energy consumption for one month of World cup 1998 traces, with respect to server provisioning scenarios. The energy-saving percentages are with respect to CPSP.

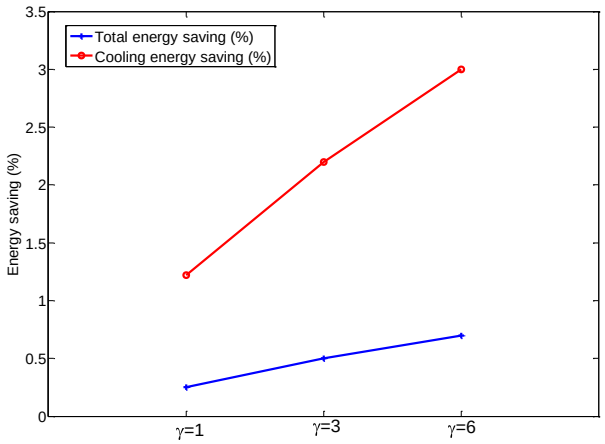

Figure 15: Energy saving of TAWD with respect to LB. The TAWD's skewing of workload toward the recirculation-efficient servers has a beneficial effect on the energy consumption.

The magnitude of energy saving of TAWD is clearly affected by the estimation of Λ^{peak}; the higher overestimation of the number of active servers, the more opportunity of energy saving using TAWD. All server provisioning schemes have to overestimate the number of active servers and leave margins to ensure QoS; these margins are used for thermal-aware workload distribution. It can be seen in Fig. 15 that TAWD can save energy up to 1% for the total energy and 3% for the cooling energy with respect to γ. Therefore, some of the energy savings of TASP which might have been sacrificed to avoid SLA violations can be compensated by the TAWD algorithm. TAWD is expected to yield higher savings when γ is higher (i.e. when the active server set is overestimated), and when the average utilization of an epoch is comparable to the epoch's peak.

7. CONCLUSIONS

This paper proposes a two-tier approach of thermal aware workload placement which leads up to 9.3% energy saving. The energy saving is directly affected by two important characteristics of Internet data centers: (i) The fluctuating nature of the Internet traffic which results in high ratio of peak over average traffic. This characteristic is a motivation of several recent research works to save energy for Internet data centers. (ii) Heat recirculation in the data center room. While recent researches have proved and introduced methods to take advantage of this characteristic for HPC data center energy saving, we adapted it to Internet data center, and show that it is possible to save more energy by taking the advantage of both characteristics.

The energy savings resulted from thermal aware server provisioning are considerable. In thermal aware workload placement, the savings come from three factors: (i) turning off unnecessary servers, (ii) selecting of the most thermally-efficient servers in the active set (TASP), and (iii) skewing the workload dispatching toward the thermally efficient servers, among the active ones (TAWD with TASP)). Based on our results TASP/TAWD (using CP-sLRH approach) achieves additional savings of 5%-8.5% with respect to a conventional active server set provisioning scheme and load balancing.

The fluctuating and stochastic nature of the Internet traffic, forces algorithm designers to overestimate the number of active servers to ensure QoS. The energy inefficiency of this overestimation can be compensated by TAWD methodology which can be used with the other power controls such as DVFS.

The performance of TASP depends upon the active server set size; it saves more energy when the active server size is smaller. Among TASP approaches, MiniMax always perform better than others. However its use in very large scale data center may be impractical due to its high complexity.

The SLA violations which may happen during an epoch results from the underestimation of peak traffic, and not from the specifics of choosing a server over another into the active set. Therefore, the extra energy savings of TASP compared to conventional CPSP is very important and is a motivation for future work to enhance both the thermal-aware server provisioning as well as data center thermal modeling. Data center management techniques such the ones discussed in this paper will be included and tested in the BlueTool[6] research infrastructure, a project that aims to provide a testbed for developing and evaluating data center management software.

8. ACKNOWLEDGMENTS

The authors thank to Edwin Verplanke (Intel Corp.) for providing the computing equipment used for §3.4.1 and §6.1.2. Further, the authors thank Tridib Mukherjee (Impact Lab) for his comments and suggestions.

9. REFERENCES

[1] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, , and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA: USENIX Association, 2008, pp. 337–350.

[2] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. K. S. Gupta, and S. Rungta, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, June 2009. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2009.06.008

[3] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling "cool": temperature-aware workload placement in data centers," in *ATEC '05: Proceedings of the annual conference on USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2005, pp. 5–5.

[4] J. Chase, D. Anderson, P. Thakar, A. Vahdat, and R. Doyle, "Managing energy and server resources in hosting centers," in *SOSP '01: Proceedings of the eighteenth ACM symposium on Operating systems principles*. New York, NY, USA: ACM, 2001, pp. 103–116.

[5] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, , and N. Gautam, "Managing server energy and operational costs in hosting centers," *SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 303–314, 2005.

[6] L. Zhang and D. Ardagna, "SLA based profit optimization in autonomic computing systems," pp. 173–182, 2004.

[7] M. Elnozahy, M. Kistler, and R. Rajamony, "Energy conservation policies for web servers," in *USITS'03: Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems*. Berkeley, CA, USA: USENIX Association, 2003, pp. 8–8.

[8] P. Ranganathan, P. Leech, D. Irwin, and J. Chase, "Ensemble-level power management for dense blade servers," in *Computer Architecture, 2006. ISCA '06. 33rd International Symposium on*, 0-0 2006, pp. 66–77.

[9] P. Bohrer, E. N. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, C. McDowell, and R. Rajamony, "The case for power management in web servers," pp. 261–289, 2002.

[10] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, pp. 1–15, 2009.

[11] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 11, pp. 1458–1472, 2008.

[12] A. M. amd Jin T., "Workload characterization of the 1998 world," Hewlett-Packard Labs, Tech. Rep., Sept. 1999.

[13] R. Sullivan and K. G. Brill, "Cooling techniques that meet "24 by forever" demands of your data center," Uptime Institute, Inc., Tech. Rep., jan 2006.

[14] T. Brunschwiler, B. Smith, E. Ruetscheo, and B. Michel, "Toward zero-emission data centers through direct reuse of thermal energy," *IBM Journal of Research and Development*, vol. 53, no. 3, pp. 11:1–11:13, 2009.

[15] R. Mullins, "HP service helps keep data centers cool," http://www.pcworld.com/article/135052/hp_service_helps_keep_data_centers_cool.html, 2007.

[16] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," *SIGMETRICS Perform. Eval. Rev.*, vol. 26, no. 1, pp. 151–160, 1998.

[17] J. Moore, J. Chase, and P. Ranganathan, "Weatherman: Automated, online, and predictive thermal mapping and management for data centers," in *IEEE International Conference on Autonomic Computing (ICAC)*, jun 2006, pp. 155–164.

[18] C. Bash and G. Forman, "Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center," HP Laboratories Palo Alto, Tech. Rep. HPL-2007-62, aug 2007.

[19] T. Mukherjee, G. Varsamopoulos, S. Gupta, and S. Rungta, "Measurement-based power profiling of data center equipment," in *IEEE International Conference on Cluster Computing.*, Sept 2007, pp. 476–477.

[20] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of Internet content delivery systems," *ACM SIGOPS Operating Systems Review*, pp. 315–327, 2002.

[21] R. D. C. Monteiro and I. Adler, "Interior path following primal-dual algorithms. part II: Convext quadratic porgramming," *Mathematical Programming*, vol. 44, pp. 43–66, 1989.

[22] Q. Tang, "Thermal-aware scheduling in environmentally coupled cyber-physical," Ph.D. dissertation, Arizona State University, August 2008.

[23] [Online]. Available: http://www.spec.org/web2009/

[6] `http://impact.asu.edu/BlueTool/`