A Cyber-Physical Systems Approach to Energy Management in Data Centers

Luca Parolini[†], Niraj Tolia[‡], Bruno Sinopoli[†], Bruce H. Krogh[†]

[†]Dept. of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
{lparolin|brunos|krogh}@ece.cmu.edu

[‡]HP Labs
1501 Page Mill Road MS 1183
Palo Alto, CA 94304
niraj.tolia@hp.com

ABSTRACT

This paper presents a new control strategy for data centers that aims to optimize the trade-off between maximizing the payoff from the provided quality of computational services and minimizing energy costs for computation and cooling. The data center is modeled as two interacting dynamic networks: a computational (cyber) network representing the distribution and flow of computational tasks, and a thermal (physical) network characterizing the distribution and flow of thermal energy. To make the problem tractable, the control architecture is decomposed hierarchically according to time-scales in the thermal and computational network dynamics, and spatially, reflecting weak coupling between zones in the data center. Simulation results demonstrate the effectiveness of the proposed coordinated control strategy relative to traditional approaches in which the cyber and physical resources are controlled independently.

Categories and Subject Descriptors

C.0 [**Computer Systems Organization**]: General—*System architectures, Modeling of computer architecture*

1. INTRODUCTION

Data centers are cyber-physical systems. Energy management depends critically upon the management of both computational (cyber) resources and cooling (physical) resources. Although these two networks are connected through the generation of thermal energy by the computational network, they are normally controlled independently. Workloads are distributed to the servers to meet performance objectives under the assumption that the cooling system will remove thermal energy as required. The cooling system responds to the thermal load generated by the servers through thermostatic control. The decoupled control strategies used today do not realize the efficiencies that could be obtained through a more holistic cyber-physical system (CPS) perspective. This paper presents a control strategy that coordinates the management of the cyber and physical resources in a data center based on the coupled network model introduced in [14].

Data center power consumption has drastically increased in the past few years. According to a report of the Environmental Protection Agency (EPA) published in 2007 [29], data center peak load power consumption was 7GW in 2006 and, at the current rate, it is expected to increase up to 12GW by 2011 leading to a cost of $7.4 billion per year. Similarly, rack power consumption has increased up to 30KW [13]. At these power usage levels, powering and cooling servers, racks, and the entire data center efficiently has become a challenging problem. Monthly management cost for a 15MW facility can be as high as $5.6M [9]. Income is determined by service level agreements (SLAs) that set the price paid by users based on the quality of service (QoS) they receive. A data center's operating margin depends on the provided quality of service: higher QoS levels typically lead to higher rates that can be charged to customers. The goal of the control strategy developed in this paper is to find the best trade-off between offered QoS and data center energy cost.

Several factors make it impractical to design and implement a single centralized controller to manage all of the data center resources. These are large cyber-physical systems with hundreds of variables that can be measured and controlled. Also, the dynamics of controlled processes span over multiple scales. For example, electricity costs can fluctuate on a time scale of hours, temperatures evolve in the order of minutes, and server power states can be changed as frequent as milliseconds. Actuators differ not only in time scales, but also in the spatial areas they influence. For example, computer room air conditioner (CRAC) reference temperatures can affect the inlet air of multiple servers, while server power states affect only single servers. These facts suggest that a hierarchical distributed control approach can best exploit the different time scales and spatial bounds of the data center processes.

Three levels of modeling and control are proposed in this paper: *data center*, *zone*, and *intra-zone*. At the data center level, we are interested in the long-term behavior of the data center. The time scale (hours) is large enough to disregard thermal and computational dynamics. Electricity price, the bulk heat management, and the long-term QoS offered are the relevant variables at this level. At the zone level, the time scale ranges from a few to a dozen minutes. At this scale we are interested in optimizing the evolution of the thermal and the computational networks. Thus, the intra-zone time scale is on the order of tenths of a second to a second. At this level we are interested in optimizing the

ICCPS '10 April 13-15,2010, Stockholm, Sweden.

quality of service in a per job fashion. The dynamics at the intra-zone level are much faster than the thermal dynamic time scale, so it is reasonable at this level to disregard the coupling between the computational and the thermal network. The intra-zone control problem reduces to a standard real-time scheduling problem that can be dealt with using methods available in the literature (e.g., [16, 17, 18, 19]).

The following section reviews previous work on data center control. Section 3 presents our integrated cyber-physical data center model and the control objectives and constraints. Section 4 develops the hierarchical distributed control strategy. Simulation results for a multi-zone data center demonstrate the effectiveness of the proposed control strategy. The concluding section summarizes the contributions of this work and describes current research directions.

2. PREVIOUS WORK

There have been several studies on server and cluster power management. These studies have proposed solutions that reduce server electricity costs by adjusting power levels to track the resource demands of the workload. The techniques used include low-power states (e.g., sleep and hibernate modes), processor dynamic voltage and frequency scaling, or DVFS [5, 15, 21, 4, 23, 7] and on/off states [3] at the local server level, and resource redirection [11, 22, 24] and task scheduling [8] at the cluster level. Chen et al. [4] address the dynamic optimization problem of server provisioning and frequency control to reduce power while minimizing SLA violations. Raghavendra et al. [23] consider the interaction between multiple power management controllers at different levels of a data center, optimizing server power without accounting for its impact on the cooling facilities.

Several studies have also examined optimization of cooling power, mainly at the data center level [1, 2, 10, 12, 20]. These studies include techniques to change workload placement to reduce air conditioning costs [12], as well as techniques to dynamically vary air flows to specific locations to improve cooling efficiency [20]. Tolia et al. [28] propose unified control of server power and cooling, as proposed in this paper, but they consider only the intra-zone (blade server) level. In this paper, we extend unified control to the zone and data center levels in the context of a comprehensive model that makes it possible to exploit tradeoffs between payoffs from SLAs and energy cost.

3. PROBLEM FORMULATION

We model a data center as two coupled dynamic networks as illustrated in Fig. 1: the *computational network* and the *thermal network*. The computational network describes relationships between workload distribution and quality of service, while the thermal network describes the relationships between power consumption, heat production, and heat exchange. As distinct workloads use data center server resources differently, heterogeneous workloads can lead to different amounts of power consumption on each server. At the same time, some servers are easier to cool than others (e.g., due to their relative positions in the rack). Thus workloads not only have different server power requirements, their distribution in the data center can also significantly impact the power required to remove the generated heat. The network models developed in this section capture the implications of these complex interactions.

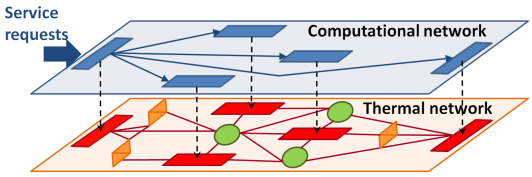

Figure 1: Data center coupled network model.

3.1 Computational Network

The computational network is a multi-class open network of queues. The data center workload is modeled as arrivals of *jobs*, defined as requests for atomic computations, where each job is a member of one of J job classes. All jobs arrive at the scheduler which routes each job to one of N computational nodes. When a job completes its execution at a computational node, it leaves the data center. We assume the time spent by each job at the scheduler is negligible.

Let t be the real variable representing time. In the proposed model $\lambda^j(t)$ represents the average (over a suitable time window) arrival rate at time t for jobs in class j. $\lambda^j(t)$ terms are stacked into the vector $\boldsymbol{\lambda}'(t) = [\lambda^1(t) \dots \lambda^J(t)]$. $s_i^j(t)$ represents the fraction of jobs in class j sent to node i by the scheduler at time t. For all $j = 1, \dots, J$ and for all t, $s_i^j(t) \in [0, 1]$ and $\sum_{i=1}^{N} s_i^j(t) = 1$. At node i, the arrival rate of jobs of class j at time t is given by $\lambda_i^j(t)$, where

$$\lambda_i^j(t) = s_i^j(t)\lambda^j(t). \tag{1}$$

$\boldsymbol{\lambda}_i(t) = \begin{bmatrix} \lambda_i^1(t) & \dots & \lambda_i^J(t) \end{bmatrix}$ is the vector of arrival rates for jobs at node i. The scheduler decisions for node i, i.e. $s_i^j(t)$, are collected in the vector

$$\boldsymbol{s}_i'(t) = \begin{bmatrix} s_i^1(t) & \dots & s_i^J(t) \end{bmatrix},$$

while the scheduler decision for all nodes are collected in the matrix $S(t)$,

$$S(t) = \begin{bmatrix} \boldsymbol{s}_1'(t) \\ \vdots \\ \boldsymbol{s}_N'(t) \end{bmatrix}.$$

Each computational node is itself a multi-class open queuing network. The way a node executes the assigned jobs determines both the QoS per job class at the node and the amount of electrical power that the node will require. Let $p_i^j(t)$ represent the ratio between the amount of computational resources used by node i to execute jobs in class j at time t and the total computational resources available at the node. A node can dedicate a certain amount of computational resources for a job class, but not more than the maximum amount of computational resources available and also, the sum of all dedicated resources cannot exceed the maximum amount of available computational resources at the node. For all t, $i = 1, \dots, N$ and $j = 1, \dots, J$, the following constraints hold

$$0 \le p_i^j(t) \le 1,$$

$$\sum_{j=1}^{J} p_i^j(t) \le 1.$$

$p_i^j(t)$ values can be grouped into the vector

$$\boldsymbol{p}_i'(t) = [p_i^1(t) \dots p_i^J(t)].$$

In the rest of the paper we shall denote as $\boldsymbol{p}_i(t)$ the vector of power states of node i and define $P(t)$ as the $N \times J$ matrix of all power states in the computational network,

$$P(t) = \begin{bmatrix} \boldsymbol{p}'_1(t) \\ \vdots \\ \boldsymbol{p}'_N(t) \end{bmatrix}.$$

Each $\boldsymbol{p}_i(t)$ vector can be regarded as an abstraction of the CPU power state concept.

Let $l_i^j(t)$ be the number of jobs of class j at the node i at time t. We consider the case where nodes can exchange jobs before they have been completed. Let $\delta_{i,z}^j(t)$ represent the number of class j jobs that move from node z to node i at time t. We assume the time spent to move jobs from one node to another is negligible. $\delta_{z,z}^j(t) = 0$, $0 \leq \delta_{i,z}^j(t) \leq l_z^j(t)$, and $\sum_{i=1}^N \delta_{i,z}^j(t) \leq l_z^j(t)$ for all $z = 1, \ldots, N$, $j = 1, \ldots, J$, and for all t. These constraints state that a node cannot exchange jobs with itself and it cannot exchange with other nodes more jobs than those available in it.

$\delta_{i,z}^j(t)$ values can be collected into a $N \times N$ matrix $\Delta^j(t)$, $[\Delta^j(t)]_{i,z} = \delta_{i,z}^j(t)$. $\Delta^1(t), \ldots, \Delta^J(t)$ are controllable variables and they represent the workload consolidation action in a data center.

The buffer evolution at a node depends on a complex relationship between job arrival at the data center, load balancing control action, job exchange between each node couple, and job execution at each server. For simple cases, like the one described in Sec. 5, the buffer evolution function can be derived by analyzing the computational network, but in general, this function will have to be estimated from data collected in the data center.

Let $q_i^j(t)$ denote the quality of service at time t for jobs in class j at node i. $\boldsymbol{q}_i(t) = \begin{bmatrix} q_i^1(t) & \ldots & q_i^J(t) \end{bmatrix}$ is the vector of QoS at time t at node i, given by

$$\boldsymbol{q}_i(t) = f_q(i, \boldsymbol{\lambda}_i(t), \boldsymbol{l}_i(t), \boldsymbol{p}_i(t)). \tag{2}$$

Let $\mathbf{c_q}(i, \boldsymbol{q}_i(t))$ represent the QoS cost vector at node i. Define $\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(t))$ as the j^{th} element of $\mathbf{c_q}(i, \boldsymbol{q}_i(t))$. We consider the case where customers pay for data center usage based on the quality of the service they receive. If the quality of service is below a certain threshold, users may be refunded by the data center[1]. If $\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(t)) < 0$ then jobs in class j executed at node i induce a payoff for the data center, while if $\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(t)) \geq 0$ execution of class j jobs at node i induce an economic loss for the data center.

Power consumption of a computation at node i at time t is denoted by $\mathsf{p}_i(t)$ and its value is given by

$$\mathsf{p}_i(t) = f_{\mathsf{p},S}(i, \boldsymbol{\lambda}_i(t), \boldsymbol{l}_i(t), \boldsymbol{p}_i(t)). \tag{3}$$

3.2 Thermal Network

The thermal network models heat generation and exchange in the data center. It is composed of three different node classes: *server* nodes, *CRAC* nodes, and *environment* nodes.

Thermal server nodes are the physical counterpart of computational server node nodes; they model how servers transform the consumed power into heat. CRAC nodes model components of the cooling subsystem; their model has been developed considering the important features of CRAC units. Although liquid cooling techniques are currently being developed and evaluated [25], the majority of data center cooling systems still rely on air cooling. Environment nodes represent those devices that cannot be used to perform computational work and that do not belong to the cooling subsystem, but that take part in the heat exchange in the data center. Devices that can be modeled as environment nodes include uninterruptible power supplies (UPS), network switches, and the external weather.

N, C, E represent the numbers of server nodes, CRAC nodes, and environment nodes, respectively. Nodes are ordered as follows: $1, \ldots, N$ are server nodes, $N+1, \ldots, N+C$ are CRAC nodes, and $N+C+1, \ldots, N+C+E$ are environment nodes.

Thermal constraints in a data center are generally expressed in terms of the inlet air temperature and humidity of each of its devices. However, since CRAC units provide automatic humidity control, only the thermal constraints need to be considered [6]. The thermal network describes only the inlet and outlet air temperatures of devices, disregarding the temperature of their internal components. $T_{\text{in}\,i}(t)$ represents the inlet air temperature of the device modeled by node i at time t, while $T_{\text{out}\,i}(t)$ represents the outlet air temperature of the device modeled by node i at time t. Input temperatures of thermal nodes are collected in the $(N+C+E) \times 1$ vector $\mathbf{T}_{\text{in}}(t)$ and output temperatures of thermal nodes are collected in the $(N+C+E) \times 1$ vector $\mathbf{T}_{\text{out}}(t)$. Temperature constraints are given as

$$\underline{\mathbf{T}_{\text{in}}} \leq \mathbf{T}_{\text{in}}(t) \leq \overline{\mathbf{T}_{\text{in}}}, \tag{4}$$

where the inequalities are meant component-wise.

We assume heat exchange in a data center is mainly due to convection; conduction and radiation are neglected. We also assume the inlet air temperature of a device can then be approximated by a linear combination of the outlet air temperatures of all other devices [27]. The coefficient relating the outlet air temperature of a device d with the inlet air temperature of another device d' depends on the amount of air that moves from d to d'. Airflow in a data center can be modeled in detail using complex computational fluid dynamics (CFD) models. To compute control strategies, we use a simplified model where air flows depend only the ON/OFF states of CRAC devices. The ON/OFF state of each CRAC node is represented by the binary variable $c_i(t)$. If $c_i(t) = 1$, then CRAC node i is in the ON state at time t, while, if $c_i(t) = 0$, the CRAC node i is in the OFF state at time t. $c_i(t)$ values are collected in the vector $\boldsymbol{c}(t)$ which takes values in $\{0,1\}^C$. The relationship between $\mathbf{T}_{\text{out}}(t)$ and $\mathbf{T}_{\text{in}}(t)$ can now be expressed via the matrix $A(\boldsymbol{c}(t))$ as

$$\mathbf{T}_{\text{in}}(t) = A(\boldsymbol{c}(t))\mathbf{T}_{\text{out}}(t). \tag{5}$$

Values of each matrix $A(\boldsymbol{x})$, $\boldsymbol{x} \in \{0,1\}^C$, can be estimated from sensor measurements following the procedure discussed in [27].

$\mathsf{p}_i(t)$ is the power consumption at time t of the i^{th} thermal node. Total data center power consumption is given by the sum of each thermal node power consumption.

Thermal Server Nodes. The results of experiments performed on a desktop machine, shown in Fig. 2, suggest that the outlet air temperature of a real server can be well approximated by the following linear different equation:

$$\dot{T}_{\text{out}\,i}(t) = k_i(T_{\text{in}\,i}(t) - T_{\text{out}\,i}(t)) + c_i \mathsf{p}_i(t), \tag{6}$$

[1]For example, the Google Apps Service Level Agreement provides for a period of free usage for a loss in QoS (http://www.google.com/apps/intl/en/terms/sla.html).

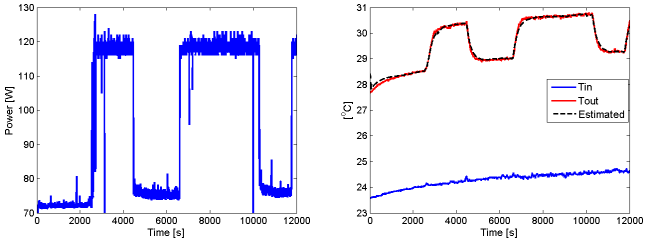

Figure 2: Power consumption, inlet, outlet, and estimated air temperature values of a desktop machine.

where i is the index of the thermal server node representing the real server, and k_i and c_i are appropriate coefficients.

CRAC nodes. A CRAC node models the cooling device (e.g., a CRAC unit) and its co-located controller. The input of the co-located controller is the reference temperature for the outlet air of the cooling device, denoted by $T_{\text{ref}_i}(t)$.

We consider a simplified case where the CRAC units can be turned on or off independently from each other and their supplied air temperature values can be controlled over a predefined interval. However, the model and the control technique developed in this paper can accommodate a more detailed description of the data center cooling subsystem.

When $c_i(t) = 1$ and $T_{\text{ref}_i}(t) < T_{\text{in}_i}(t)$ then $T_{\text{out}_i}(t)$ tends to the reference temperature $T_{\text{ref}_i}(t)$, while $T_{\text{out}_i}(t)$ tends to $T_{\text{in}_i}(t)$ when $T_{\text{ref}_i}(t) \geq T_{\text{in}_i}(t)$. When $c_i(t) = 0$ then $T_{\text{out}_i}(t)$ tends to $T_{\text{in}_i}(t)$. We consider the following relationship between the reference and the output temperature of a CRAC node i

$$\begin{aligned}\dot{T}_{\text{out}_i}(t) = -k_i T_{\text{out}_i}(t) + (1 - c_i(t))k_i T_{\text{in}_i}(t) + \\ + c_i(t) k_i \min\{T_{\text{ref}_i}(t), T_{\text{in}_i}(t)\},\end{aligned} \tag{7}$$

where $\frac{1}{k_i} > 0$ represents the time constant of the CRAC node.

Power consumption of a CRAC node is given by

$$\mathsf{p}_i(t) = f_{\mathsf{p},C}(i, \boldsymbol{c}(t), T_{\text{in}_i}(t), T_{\text{ref}_i}(t), T_{\text{out}_i}(t)), \tag{8}$$

where $f_{\mathsf{p},C}$ accounts for the coefficient of performance (COP) of each CRAC unit modeled as a CRAC node, and the additional components that consume energy in a CRAC unit even when no power is required for cooling the air.

Environment Nodes. Environment nodes represent the data center devices that cannot be used to perform computations or to control the environment, but that nonetheless take part to the heat exchange. Nodes in this category can be represented using two different models. The first model assumes that the output temperature of the node is solely a function of time, power consumption is always zero and the input temperature has no relationship with the output temperature:

$$\begin{cases} T_{\text{out}_i}(t) = f_{T_{\text{out}},E}(i,t) \\ \mathsf{p}_i(t) = 0 \end{cases}. \tag{9}$$

This kind of model can be used to describe the effect of the external weather on the data center.

The second model uses a relationship between power consumption, input temperature, and output temperature similar to the thermal server node case:

$$\dot{T}_{\text{out}_i}(t) = k_i(T_{\text{in}_i}(t) - T_{\text{out}_i}(t)) + c_i \mathsf{p}_i(t), \tag{10}$$

where $\mathsf{p}_i(t)$ can either be constant, or function of other data center variables. For example, if we consider the model of a UPS unit, then modeling $\mathsf{p}_i(t)$ being proportional to the total server power consumption can be a reasonable assumption. If instead, we want to model a network switch, then a constant power consumption model can be a better approximation of the real device power consumption.

We will denote as E_1 the number of environment nodes described by (9) and as E_2 ones modeled by (10).

3.3 Network coupling

Computational and thermal networks are coupled via the computational node power consumption. Each computational node is uniquely associated to a thermal server node. Power consumption of computational node i becomes one of the inputs that control the output temperature evolution of the i^{th} thermal server node

$$\begin{aligned}\dot{T}_{\text{out}_i}(t) = k_i(T_{\text{in}_i}(t) - T_{\text{out}_i}(t)) + \\ + c_i f_{\mathsf{p},S}(i, \boldsymbol{\lambda}_i(t), \boldsymbol{l}_i(t), \boldsymbol{p}_i(t)).\end{aligned} \tag{11}$$

Eq.(11) models the relationship between workload execution and outlet air temperature evolution in a server. We use the term *server node* to refer to the combination of a computational server node and a thermal server node.

3.4 Control Objective

Let $\mathsf{c}_\mathsf{p}(t)$ denote the electricity price at time t. As shown in Fig. 3, electricity price fluctuates significantly over time. This fluctuation has to be considered in order to determine the best trade-off between increasing the offered QoS and decreasing the cost of powering the data center. The data center operating cost at time t is given by

$$\mathsf{c}_\mathsf{p}(t)||\mathbf{p}(t)||_1 + \sum_{i=1}^{N}\sum_{j=1}^{J} s_i^j(t) \mathsf{c}_q{}^j(i, \boldsymbol{q}_i(t)),$$

where $||\mathbf{p}(t)||_1$ is the sum of the power consumption of all thermal nodes, while $s_i^j(t)$ terms are used to scale the quality of service cost obtained at node i for jobs in class j by the ratio of class j jobs sent to node i at time t.

Define $\boldsymbol{u}(t)$ as the vector of all controllable variables, i.e.

$$\boldsymbol{u}(t) = \begin{bmatrix} vec(S(t)) \\ vec(P(t)) \\ vec(\Delta^1(t)) \\ \vdots \\ vec(\Delta^J(t)) \\ \boldsymbol{c}(t) \\ \mathbf{T}_{\text{ref}}(t) \end{bmatrix},$$

where $vec(X)$ is the vector obtained by stacking the columns of the matrix X on top of one another. Let $\boldsymbol{\beta}(t)$ be the collection of vectors $\mathbf{p}_{E_1}(t)$, $\mathbf{T}_{\text{out}\,E_2}(t)$ and $\boldsymbol{\lambda}(t)$. $\boldsymbol{\beta}(t)$ represents the vector of uncontrollable inputs and we assume its value to be known at every instant t.

The goal of a data center controller is to design the optimal control $\boldsymbol{u}(\tau)$, $\tau \in [t_0, t_0 + \mathcal{T}]$, to minimize the following cost

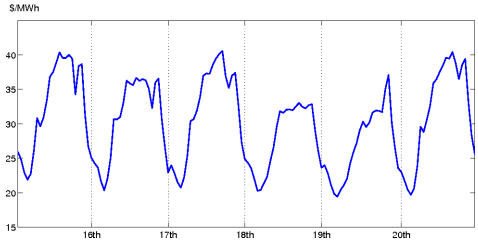

Figure 3: Electricity day-ahead price in northern west Pennsylvania in Sept. 2009 (from http://www.pjm.com/home.aspx).

function

$$J(\boldsymbol{u};\boldsymbol{\beta},t_0,\mathcal{T}) = \int_{t_0}^{t_0+\mathcal{T}} \mathsf{c_p}(\tau)||\mathbf{p}(\tau)||_1 + \\ + \sum_{i=1}^{N}\sum_{j=1}^{J} s_i^j(t)\mathsf{c_q}^j(i,\boldsymbol{q}_i(\tau))d\tau, \quad (12)$$

subject to the thermal and computational constraints defined above. In (12) t_0 is the initial time of the optimization problem and $\mathcal{T} > 0$ is the optimization time horizon.

The large number of variables and the (generally) nonlinear constraints that govern computational and the thermal networks imply that in most cases (12) cannot be optimized directly. However, the natural data center modularity and the different time scales at which different processes evolve can be exploited using reduced-order models and relying on a hierarchical distributed control approach.

4. CONTROL STRATEGY

Two hierarchy levels are considered in this paper: the *data center level* and the *zone level*. The control decisions and manipulated variables used at each level reflect the dominant dynamics of the time scale being addressed at each level, with slower thermal dynamics being most relevant at the higher levels and fast dynamics of the computational systems dominating the lower levels.

The data center level deals with the bulk management of workload and thermal management, using workload projections at the hourly and daily levels to schedule cooling and make major decisions about task and data allocations. The zone level concerns the allocation of workload and cooling in sub-areas of the data center and acts on a scale of minutes.

A discrete-time optimal control approach is considered at each level of the hierarchy. $\tau_\Delta(\nu)$ represents the controller sampling time at level ν, where level $\nu = 0$ corresponds to the data center level and $\nu = 1$ to the zone level. We assume $0 < \tau_\Delta(1) \ll \tau_\Delta(0)$. Continuous time constraints defined in Sec. 3 are discretized according to each hierarchy level sampling time, and controllable variables are considered constant between sampling intervals.

In the following section we introduce some additional assumptions to simplify the description of the control approach.

4.1 Aggregation and disaggregation

The idea of grouping multiple heterogeneous nodes into single server nodes stems from the analysis of the following real data center cases: containerized data centers (e.g., the Sun Modular Data center[2]), row- and rack-oriented cooling [13], and blade server architectures. When the optimization of a single container, row, rack, or blade enclosure is the goal of the control problem, we can model these units of equipment as the coupling of a computational network and a thermal network. In each a case, the network nodes correspond to the detailed behaviors of the components internal to the units of equipment. Alternatively, when the goal of the control problem is the optimization of the whole data center, each unit of equipment can be modeled as a single server node representing the aggregate behavior of its internal components.

Aggregating multiple heterogeneous devices into a single server node may be a coarse approximation to the actual behavior of a complex unit of equipment, and there is no general method to obtain the "best" aggregate model for complex nonlinear models like those developed in Sec. 3. The aggregation of multiple nodes at one level into single nodes at the level above will have to be based on engineering insight and data-driven analysis. Minimizing the differences between the aggregate evolutions of devices modeled as single nodes at one level and the evolution of the sets of lower-level nodes is the duty of controllers at lower levels in the hierarchy. In this paper we consider the aggregation of server nodes only.

The disaggregation problem is to define a function to associate input, output, and state variables of an aggregate node at one level to input, output, and state variables of the associated set of nodes in the next lower level in the hierarchy. The choice of the best disaggregation function is delegated to the lower-level controllers. In the proposed hierarchical control strategy, the desired aggregated state variables, i.e., server power states, output temperatures, and buffer length, are not transmitted to the lower-level controllers. The top-down communication consist only of the desired power cost and QoS cost. The goal of the zone-level controller is to find the best disaggregation function that minimizes both the difference between the total cost of power consumption and the desired (aggregated) cost of power consumption and the difference between the total QoS cost and the desired (aggregated) QoS cost.

4.2 Data center level

Consider a computational network of N nodes and assume the job arrival rate is the realization of J independent Poisson processes, each having parameter $\lambda^j(t)$. The Poisson process assumption is used then to derive the equations relating job arrival rate and power state vector of a node to the expected buffer length. In general, however, we expect these relationship to be learned from past data. We approximate the job expected arrival rate values in the k^{th} interval with a constant value denoted as $\lambda^j(k)$.

At the data center level each server node, i.e., each combination of a computational server node and a thermal server node, represents an aggregation of multiple server nodes controlled by a zone-level controller.

The scheduler enforces a random policy based on the $s_i^j(k)$ values, i.e., the probability to send an incoming job of class j during the k^{th} interval to node i is $s_i^j(k)$. The expected arrival rate at node i for jobs in class j can be approximated as $\lambda_i^j(k) = \lambda^j(k)s_i^j(k)$. Job execution time at each node i is

[2]http://www.sun.com/products/sunmd/s20/

exponentially distributed with parameter $\mu_i^j(k) = \overline{\mu}_i^j p_i^j(k)$, where $\overline{\mu}_i^j \geq 0$ is the maximum job execution rate for job in class j at node i. Computational nodes in this case are just a collection of J M/M/1 queues [26].

Due to the large sampling period used at this level, we assume that we can treat the computational nodes as if they were at their invariant distribution (when it exists) and approximate average values of their variables with their expected values.

The expected buffer length of jobs in class j at node i is given by

$$\hat{l}_i^j(k) = \begin{cases} \dfrac{\lambda_i^j(k)}{\mu_i^j(k) - \lambda_i^j(k)} & \mu_i^j(k) > \lambda_i^j(k) \\ +\infty & \text{otherwise} \end{cases}. \quad (13)$$

Since $\hat{l}_i^j(k)$ is independent of $\hat{l}_i^j(k-1)$, we neglect the possibility to move jobs from one computational node to another, i.e., $\Delta^j(k) = 0$ for all $j = 1, \ldots, J$ and $k \in \mathbb{Z}_0$.

We define quality of service obtained at node i for jobs in class j as

$$q_i^j(k) = \mu_i^j(k) - \lambda_i^j(k). \quad (14)$$

In this case the chosen QoS does not depend on $l_i^j(k)$.

Let $\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(k))$ be the j^{th} element of $\mathbf{c_q}(i, \boldsymbol{q}_i(k))$. We define

$$\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(k)) = \begin{cases} +\infty & q_i^j(k) \leq 0 \\ \overline{\mathsf{c}} & 0 < q_i^j(k) \leq \underline{q} \\ \dfrac{\alpha}{q_i^j(k)} + \underline{\mathsf{c}} & q_i^j(k) > \underline{q} \end{cases}, \quad (15)$$

where α satisfies $\alpha/\underline{q} + \underline{\mathsf{c}} = \overline{\mathsf{c}}$, while $\underline{\mathsf{c}}$ and $\overline{\mathsf{c}}$ represent respectively the lower and the upper bound of the QoS cost function. In order to avoid the case $\mu_i^j(k) \leq \lambda_i^j(k)$ we set $\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(k))$ to infinity when $q_i^j(k) \leq 0$. In this particular case quality of service of all job classes is bounded in the same intervals $[\underline{\mathsf{c}}, \overline{\mathsf{c}}]$, and also, the cost function of jobs in class j depends solely on the quality of service achieved for class j jobs. In general, however, it may be useful to consider different bounds for different job classes, or more complex relationships between QoS values and the induced cost function.

Let $\mathsf{p}_i(k)$ represent the average power consumption of devices modeled by thermal node i during the k^{th} interval. Thermal server node power consumption is the sum of its static ($\mathsf{p}_{S,i}(k)$) and dynamic components ($\sum_{j=1}^J \mathsf{p}_{D,i}^j(k)$),

$$\mathsf{p}_i(k) = \mathsf{p}_{S,i}(k) + \sum_{j=1}^{J} \mathsf{p}_{D,i}^j(k). \quad (16)$$

Static power consumption is given by

$$\mathsf{p}_{S,i}(k) = \overline{\mathsf{p}}_{S,i} \sum_{j=1}^{J} p_i^j(k),$$

where $\overline{\mathsf{p}}_{S,i}$ represents the maximum static power consumption of node i. The dynamic part of the power consumption is

$$\mathsf{p}_{D,i}^j(k) = \begin{cases} \overline{\mathsf{p}}_{D,i}^j p_i^j(k) \dfrac{\lambda_i^j(k)}{\mu_i^j(k)} & \mu_i^j(k) > \lambda_i^j(k) \\ \overline{\mathsf{p}}_{D,i}^j p_i^j(k) & \text{otherwise} \end{cases},$$

where $\overline{\mathsf{p}}_{D,i}^j$ is the maximum dynamic power consumption due to the class j jobs. $\frac{\lambda_i^j(k)}{\mu_i^j(k)}$ approximates the ratio of the time interval spent by node i executing jobs in class j during the k^{th} interval. Since ergodicity is not guaranteed, this approximation may be coarse.

For the thermal network we assume the sampling time at this level is large enough so that the devices represented by thermal nodes can reach thermal equilibrium in a time period much shorter than the sampling time. Eq.(6) then, can be rewritten as

$$T_{\text{out}\,i}(k) = T_{\text{in}\,i}(k) + \frac{c_i}{k_i} \mathsf{p}_i(k), \quad (17)$$

for all $i = 1, \ldots, N$ with $\mathsf{p}_i(k)$ given by (16).

The time constant $\frac{1}{k_i}$ of each CRAC node is assumed to be significantly smaller than the sampling period $\tau_\Delta(0)$. We can then assume

$$T_{\text{out}\,i}(k) = \min\{T_{\text{ref}\,i}(k), T_{\text{in}\,i}(k)\}$$

for all $k \in \mathbb{Z}_0$ and $i = N+1, \ldots, N+C$. As discussed in [12], average power consumption of CRAC node i can therefore be approximated as

$$\mathsf{p}_i(k) = c_i(k) \Big[\max \Big\{ 0, f_i(\boldsymbol{c}(k)) c_p \frac{T_{\text{in}\,i}(k) - T_{\text{ref}\,i}(k)}{COP_i(T_{\text{ref}\,i}(k))} \Big\} + \\ + \mathsf{p}_{f,i}(\boldsymbol{c}(k)) \Big]$$

where $f_i(\boldsymbol{c}(k))$ is the average air mass that flows through CRAC unit i in the k^{th} interval when the CRAC cooling state is $\boldsymbol{c}(k)$. c_p is the specific heat of the air, COP_i is the COP of node i, and $\mathsf{p}_{f,i}(\boldsymbol{c}(k))$ is the fan power of the CRAC when the cooling state is $\boldsymbol{c}(k)$.

Environment nodes modeled by (10), e.g., switches and UPS units, are approximated by (17), while those environment node modeled by (9), e.g., external environment temperatures, are approximated as

$$\begin{cases} T_{\text{out}\,i}(k) = f_{T_{\text{out}},E}(i,k) \\ \mathsf{p}_i(k) = 0 \end{cases}. \quad (18)$$

The input temperatures of all thermal nodes are given by

$$\mathbf{T}_{\text{in}}(k) = A(\boldsymbol{c}(k)) \mathbf{T}_{\text{out}}(k). \quad (19)$$

Controller formulation: Let $\mathcal{T}_0$ be the horizon for the optimization problem. The control variables at time k are given by: $P(k), S(k), \mathbf{T}_{\text{ref}}(k)$, and $\boldsymbol{c}(k)$, where $\mathbf{T}_{\text{ref}}(k) = [T_{\text{ref}\,N+1}(k) \ldots T_{\text{ref}\,N+C}]$ is the vector of reference temperature at time k. We define $\boldsymbol{u}$ as the collections of such controllable variables in the discrete time interval $\{k, \ldots, k+\mathcal{T}_0\}$.

Denote with $\mathbf{p}_{E_1}(k)$ the power consumption at time k of the environment nodes modeled by (9) and with $\mathbf{T}_{\text{out}\,E_2}(k)$ the output temperature of the environment node at time k modeled by (10). Define $\beta(k)$ as the collection of vectors $\mathbf{p}_{E_1}(k), \mathbf{T}_{\text{out}\,E_2}(k)$, and $\boldsymbol{\lambda}(k)$. We assume $\beta(k)$ to be completely known at any time k. $\boldsymbol{\beta} = \big[\beta(k), \ldots, \beta(k+\mathcal{T}_0)\big]$ is the set of uncontrollable inputs that affect the optimization function.

The data center cost function can then be written as

$$J(\boldsymbol{u}; \boldsymbol{\beta}, k, \mathcal{T}_0) = \sum_{r=k}^{k+\mathcal{T}_0} \Big(\mathsf{c}_\mathsf{p}(r) ||\mathbf{p}(r)||_1 + \\ + \sum_{i=1}^{N} \sum_{j=1}^{J} s_i^j(r) \mathsf{c}_q{}^j(i, \boldsymbol{q}_i(r)) \Big). \quad (20)$$

Constraints for this level depend only on the current time. Therefore, a simple optimization problem that disregards future values of the inputs can be formulated, i.e., $\mathcal{T}_0 = 0$. Additional costs can be considered in the cost function in order to reduce the variation of optimal control variables between time instants $k-1$ and k.

The solution to the above optimization problem gives the optimal control $u^\star(k)$. The optimal CRAC power state $\boldsymbol{c}^\star(k)$ and reference temperature vector $\mathbf{T}_{\text{ref}}{}^\star(k)$ are sent to the controllers of the CRAC units, while the switching matrix $S^\star(k)$ is used to tune the scheduler. Values of the $P^\star(k)$ matrix are used jointly with other optimal control variables to derive the optimal desired QoS cost and the power consumption cost for each of the server nodes.

The optimization problem formulated at this level will in general be a mixed-integer, nonlinear program (MINLP), where the integer constraint is due to the CRAC cooling state $\boldsymbol{c}(k)$.

4.3 Zone level

The decision variables for this level are the load balancing at the scheduler, the power state of each computational node and the job exchange among computational nodes. Constraints related to the load balancing and power state variables are the same as the ones discussed in the previous section.

Buffer evolution at this level has to be considered. The expected value of $l_i^j(k+h)$, $h \geq 0$, is computed as

$$\hat{l}_i^j(k+h) = \sum_{n=0}^{+\infty} p_{i,n}^j(k+h|k)n, \tag{21}$$

where $p_{i,n}^j(k+h|k)$ is the probability that $l_i^j(k+h) = n$ given the value $l_i^j(k)$ at time k. $p_{i,n}^j(k+h|k)$ values for M/M/1 queues can be found, for example, in [26]. According to (21), we have $\hat{l}_i^j(k) = l_i^j(k)$ for all $k \in \mathbb{Z}$, $i = 1, \ldots, N$ and $j = 1, \ldots, J$.

At this level we define $q_i^j(k)$ to be equal to $\hat{l}_i^j(k)$. Let $\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(k))$ be the j^{th} element of $\mathbf{c_q}(i, \boldsymbol{q}_i(k))$, we set

$$\mathsf{c}_q{}^j(i, \boldsymbol{q}_i(k)) = \begin{cases} \alpha q_i^j(k) + \bar{\mathsf{c}} & 0 \leq q_i^j(k) < \underline{q} \\ \underline{\mathsf{c}} & q_i^j(k) \geq \underline{q} \end{cases}, \tag{22}$$

where α is such that $\alpha \underline{q} + \bar{\mathsf{c}} = \underline{\mathsf{c}}$ and $\underline{\mathsf{c}} < \bar{\mathsf{c}}$.

Average node power consumption comprises both a static and a dynamic term:

$$\mathsf{p}_i(k) = \mathsf{p}_{S,i}(k) + \sum_{j=1}^{J} \mathsf{p}_{D,i}^j(k), \tag{23}$$

where the static power consumption is given by

$$\mathsf{p}_{S,i}(k) = \bar{\mathsf{p}}_{S,i} \sum_{j=1}^{J} p_i^j(k)$$

and the dynamic part of the power consumption at a server node i is given by

$$\mathsf{p}_{D,i}(k) = \begin{cases} \mathsf{p}_{D,i}^j p_i^j(k) & \hat{l}_i^j(k) > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{24}$$

The sampling time at this level is not considered large enough to be able to disregard the thermal dynamics. Thermal equations described in Sec. 3 have to be discretized and introduced as constraints.

Controller formulation: let $\mathcal{T}_1$ be the horizon for the optimization problem at the zone level. Similarly to the data center level, we denote with $\boldsymbol{u}$ the collection of controllable variables in the discrete time interval $\{k, \ldots, k+\mathcal{T}_1\}$ and with $\beta(k)$ the collection of uncontrollable inputs that affect the optimization function. In this example we consider $\beta(k)$ to be completely known at every time k.

Let $\mathsf{c}_q{}^{j\star}$ be the optimal QoS cost for class j jobs obtained for this zone at the data center level and $\mathsf{c}_\mathsf{p}^\star$ the optimal power consumption cost obtained for this zone at the data center level. All of the $\mathsf{c}_q{}^{j\star}$ variables, $j = 1, \ldots, J$ and $\mathsf{c}_\mathsf{p}^\star$ are scaled in order to compensate for the different time horizons used at the data center level and at the current zone level.

The cost function considered by the controller at this level is

$$J(\boldsymbol{u}; \boldsymbol{\beta}, k, \mathcal{T}_1) = \gamma, \tag{25}$$

where γ has to enforce the following constraints

$$\begin{cases} \left(\sum_{r=k}^{k+\mathcal{T}_1} \sum_{i=1}^{N} s_i^j(r) \mathsf{c}_q{}^j(i, \boldsymbol{q}_i(r)) \right) - \mathsf{c}_q{}^{j\star} \leq \gamma \quad j = 1, \ldots, J \\ \left(\sum_{r=k}^{k+\mathcal{T}_1} \mathsf{c}_\mathsf{p}(r) ||\mathbf{p}(r)||_1 \right) - \mathsf{c}_\mathsf{p}^\star \leq \gamma. \end{cases}$$

Even though at this level we are dealing with a stochastic optimization problem (future values of the buffer length are unknown), we are able to formulate a deterministic optimization problem since all of the constraints and the objective function depend only on the expected values of the buffer lengths. Differences between the expected values of the buffer lengths and the true future values will be managed by the receding horizon controller.

5. SIMULATION RESULTS

We consider a data center composed of 6 racks of 42 1U servers each and 3 CRAC units with the configuration illustrated in Fig. 4. Racks and CRAC units are not scaled. Rectangles in Fig. 4 are only used to represents the relative positions of different components.

The server nodes are identical and the CRAC nodes are identical, both at the data center and at the zone level. The physical positions of the computational and cooling units lead to different thermal interactions among the nodes. There is a single class of jobs, i.e., $J = 1$. In this simulation study we consider the controller at the data center level. No cost is incurred for changes in the control variables from step to step and only the cost for the immediate time period are considered in the optimization, i.e., $\mathcal{T}_0 = 0$.

Our proposed control strategy is compared against two other possible control solutions: an *uncoordinated* controller that optimizes the computational and thermal networks separately, and a *uniform* strategy that uses all available computational and cooling resources to meet the workload. Results are analyzed based on the aggregated models used at the data center level.

Thermal constraints for all of the control strategies are

$$5 \leq T_{\text{in}\,i}(k) \leq 25, \tag{26}$$

for all $i = 1, \ldots, 6$, while power consumption cost is considered fixed at 3 cent/KWhr.

The uncoordinated strategy optimizes $p_i^1(k)$ and $s_i^1(k)$ values in order to minimize the cost of powering servers plus the

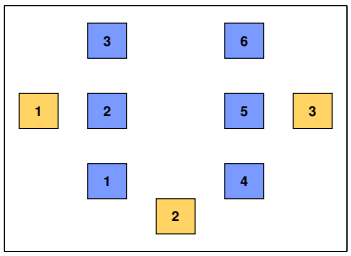

Figure 4: Data center layout. 6 racks and 3 CRAC units.

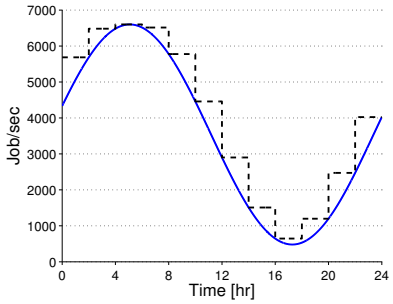

Figure 5: Average job arrival rate and its quantized version.

QoS cost, disregarding the cost of powering CRAC nodes. Once the optimal values of $p_i^1(k)$ and $s_i^1(k)$ are obtained, the uncoordinated strategy finds the best CRAC cooling state value in order to minimize the total CRAC power cost while enforcing the thermal constraints.

The uniform algorithm does not minimize a cost function and it does not consider the temperature evolution in the data center. It sets all CRAC units in the ON state, the power states of all computational nodes at 1, and distributes the workload uniformly among the servers, i.e., $s_i^1(k) = \frac{1}{N}$ for all k and $i = 1, \ldots, 6$. It sets the reference temperature of each CRAC unit to 15^oC so that, under all possible conditions, thermal constraints are enforced.

The optimization problems are solved using the TomSym[3] modeling language with KNITRO[4] as the MINLP solver.

[3]http://tomsym.com/
[4]http://www.ziena.com/knitro.htm

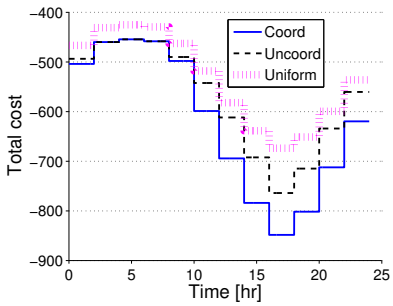

Figure 6: Total cost over time.

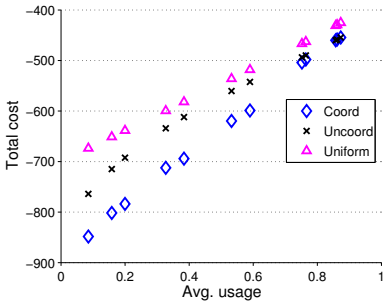

Figure 7: Total cost over average data center usage.

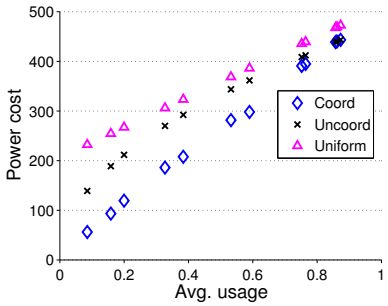

Figure 8: Power cost over average data center usage.

The average job arrival rate $\lambda^1(t)$ and the quantized version, considered by the three different data center controllers, are depicted in Fig. 5.

Define the average data center usage as

$$\frac{1}{N} \sum_{i:s_i^1 \neq 0} \frac{\lambda_i^1(k)}{\mu_i^1(k)}, \tag{27}$$

where the summation includes only computational nodes executing jobs. Compared to Fig. 5, large average data center usage values correspond to large job arrival rate values (e.g., values obtained around the 4^{th} hour), while small average data center usage values correspond to small job arrival rate values (e.g., values obtained around hour the 16^{th} hour).

As shown in Figs. 6 and 7, all three controllers increase the total cost as the average data center usage increases. Also, all of the three controllers are able to obtain negative total cost values under all of the average data center usage conditions. This implies that the policies enforced by the three controllers are always profitable. As the data center usage tends to one, the set of admissible inputs that satisfy thermal and computational constraints reduce and hence, the total cost values obtained by different controllers converge.

Figures 8 and 9 show the power cost and QoS cost, respectively, as a function of the average data center usage. The coordinated controller is able to obtain the most linear relationship between cost of power and average data center usage and in particular, it obtains the smallest power cost as the data center usage tends to zero. The uniform controller achieves the highest QoS, as it always sets server power states to one, but this also results in the largest cost of power. Unlike the uncoordinated controller, the coordinated one takes into account both the server and the CRAC node power consumptions when computing the best tradeoff

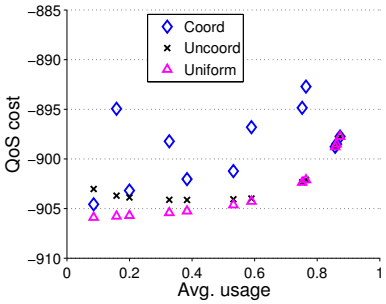

Figure 9: QoS cost over average data center usage.

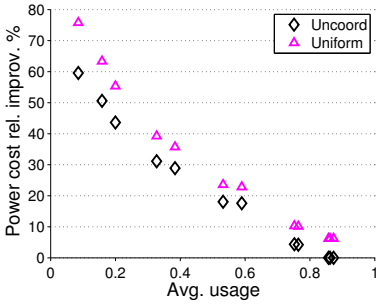

Figure 10: Relative improvement in the power cost of the coordinated strategy respect to the uncoordinated and uniform strategies.

between cost of powering and QoS cost. CRAC node power consumption is in general, a non monotonic function of the average data center usage. This explains why the QoS cost values obtained by the coordinated controller do not lie on a monotonic curve.

Figure 10 shows the percentage improvement of the power consumption cost of the coordinated strategy with respect to the uniform and the uncoordinated strategy. In this particular simulation we can see how a slight increase in the QoS cost can lead to a large reduction of the power consumption cost, particularly when the data center is under utilized. Since all server nodes and all of the CRAC nodes are identical to each other, the large difference in the power consumption cost depends on the thermal coupling between server and CRAC nodes.

Differences between coordinated and uncoordinated control policy, also depend on the particular QoS cost function chosen. Figures 11, 12, and 13 present the results for a stricter QoS cost function for which even a small reduction of the power state value leads to a large difference in the QoS cost. In this case, the coordinated controller is able to outperform the uncoordinated controller only for a few data center average utilization values.

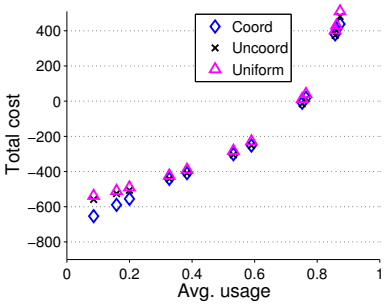

Figure 11: Total cost over time.

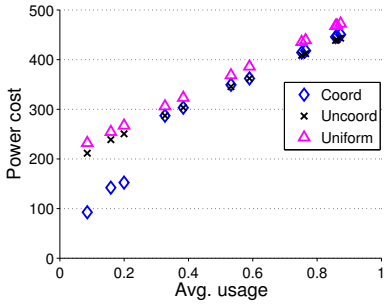

Figure 12: Power cost over average data center usage.

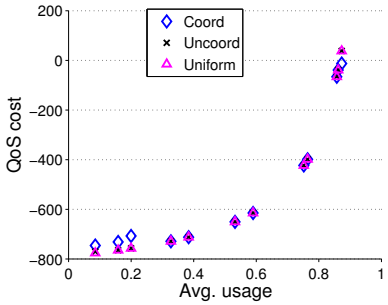

Figure 13: QoS cost over average data center usage.

6. DISCUSSION

This paper presents a control strategy for realizing best trade-off between satisfying user requests and energy consumption in a data center based on a model that includes both the cyber and physical elements in the system. Simulation results demonstrate that improvements can be realized using the proposed coordinated strategy relative to traditional approaches that manage the cyber and physical resources separately. The simulation results presented in this paper, which are the first studies of coordinated cyber-physical control at the data center level, indicate that the extent of the savings depends on many factors, including the level of the workload relative to the overall capacity of the data center. More research is needed to determine what factors are most significant in determining the effectiveness of coordinated control.

We are also pursuing several other directions of research. The proposed model is oriented toward applications where the statistics and dynamics of the workload are known. For cases where the workload characteristics are not known a priori, methods need to be developed to determine the model parameters from historical data or through real-time estimation. Other model parameters also need to be determined empirically. We are currently performing experiments to

determine thermal network parameters for the Data Center Observatory (DCO), a research data center at Carnegie Mellon.[5] In the future, we plan to implement the coordinated controllers at the data center and zone levels in the DCO.

Extensions to the proposed framework are also being considered to incorporate bi-directional communication between the levels. Currently our hierarchical distributed approach allows only for information flow from the top down: higher-level controllers provide targets for the disaggregated controllers in lower levels. Communication from the bottom up could enhance the ability to deal with modeling inaccuracies at the higher levels. For example, a zone level controller could ask the data center controller to generate a new control policy if there is insufficient capacity to achieve the requested levels of service. Extensions are also needed to incorporate additional knowledge, such as more frequent fluctuations in energy prices.

7. ACKNOWLEDGMENTS

This research was supported in part by NSF Grant ECCS-0925964 and in part by Pennsylvania Infrastructure Technology Alliance (PITA) Grant C000032167.

8. REFERENCES

[1] C. Bash and G. Forman. Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. In *USENIX Annual Technical Conference*, June 2007.

[2] C. E. Bash, C. D. Patel, and R. K. Sharma. Dynamic thermal management of air cooled data centers. In *ITHERM*, 2006.

[3] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle. Managing energy and server resources in hosting centers. In *Proc. of the 18th ACM Symposium on Operating Systems Principles*, Oct. 2001.

[4] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing server energy and operational costs in hosting centers. In *SIGMETRICS*, June 2005.

[5] E. N. Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In *PACS*, Feb. 2002.

[6] T. Evans. Humidification strategies for data centers and network rooms. Technical Report White Paper 58, APC, 2004.

[7] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *ACM SIGMETRICS*, 2009.

[8] L. Grit, D. Irwin, A. Yumerefendi, and J. Chase. Virtual machine hosting for networked clusters: Building the foundations for "autonomic" orchestration. In *VTDC*, Nov. 2006.

[9] J. Hamilton. Cost of power in large-scale data centers. `http://perspectives.mvdirona.com`, Nov. 2008.

[10] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini. Mercury and freon: Temperature emulation and management for server systems. In *ASPLOS*, Oct. 2006.

[11] T. Heath, B. Diniz, E. V. Carrera, W. M. Jr., and R. Bianchini. Energy conservation in heterogeneous server clusters. In *PPOPP*, June 2005.

[12] J. Moore, J. Chase, P. Ranganathan, R. Sharma. Making scheduling "cool": temperature-aware workload placement in data centers. In *ATEC*, 2005.

[13] K. Dunlap, N. Rasmussen. The advantages of row and rack-oriented cooling architectures for data centers. White paper, APC, 2006.

[14] L. Parolini, B. Sinopoli, B. H. Krogh. A unified thermal-computational approach to data center energy management. In *Proc. of FEBID, CPS Week*, 2009.

[15] C. Lefurgy, X. Wang, and M. Ware. Power capping: A prelude to power shifting. *Cluster Computing*, 2008.

[16] C. Lu, J. Stankovic, T. Abdelzaher, G. Tao, S. Son, and M. Marley. Performance specification and metrics for adaptive real-time systems. In *Proc. of the Realtime Systems Symposium*, Dec. 2000.

[17] Y. Lu, T. Abdelzaher, C. Lu, L. Sha, and X. Liu. Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers. In *IEEE RTAS*, 2003.

[18] Y. Lu, T. Abdelzaher, C. Lu, and G. Tao. An adaptive control framework for qos guarantees and its application to differentiated caching. In *IEEE RTAS*, 2002.

[19] J. Mao and C. Cassandras. Optimal control of multi-stage discrete event systems with real-time constraints. In *IEEE Trans. on Automatic Control*, Jan. 2009.

[20] C. D. Patel, C. E. Bash, R. Sharma, M. Beitelmam, and R. J. Friedrich. Smart cooling of data centers. In *IPACK*, 2003.

[21] T. Pering, T. Burd, and R. Brodersen. The simulation and evaluation of dynamic voltage scaling algorithms. In *SLPED*, Aug. 1998.

[22] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath. Dynamic cluster reconfiguration for power and performance. *Compilers and Operating Systems for Low Power*, 2003.

[23] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No "power" struggles: Coordinated multi-level power management for the data center. In *ASPLOS*, Mar. 2008.

[24] K. Rajamani and C. Lefurgy. On evaluating request-distribution schemes for saving energy in server clusters. In *ISPASS*, Mar. 2003.

[25] P. Rumsey. Overview of liquid cooling systems. Slides, 2007. Rumsey Engineers.

[26] T. L. Saaty. *Elements of queuing theory with applications*, chapter 4. McGraw-Hill, 1961.

[27] Q. Tang, T. Mukherjee, S. K. S. Gupta, and P. Cayton. Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters. In *ICISIP*, Oct. 2006.

[28] N. Tolia, Z. Wang, P. Ranganathan, C. Bash, M. Marwah, and X. Zhu. Unified power and cooling management in server enclosures. In *InterPACK*, 2009.

[29] U.S. Environmental Protection Agency (EPA). Report to congress on server and data center energy efficiency, public law 109-431, Aug. 2007.

[5] http://www.pdl.cmu.edu/DCO/