

# A Unified Thermal-Computational Approach to Data Center Energy Management

Luca Parolini, Bruno Sinopoli, Bruce H. Krogh  
Dept. of Electrical and Computer Engineering  
Carnegie Mellon University  
Pittsburgh, PA 15213-3890  
{lparolin|brunos|krogh}@ece.cmu.edu

## ABSTRACT

This paper presents a unified, coordinated, thermal-computational approach to the data center energy management problem. A data center is modeled as two coupled networks: a computational and a thermal network. The first network describes the computation performance of the data center, while the latter describes temperature evolution of the devices housed in the data center, based on their electrical power consumption and cooling. Server power states influence both networks. We formulate the energy control problem as a Markov decision process (MDP) and compare through an example, the performance of a controller derived using the proposed unified thermal-computational approach against a controller that disregards the coupling between the two networks. Simulation results suggest further research directions.

## Categories and Subject Descriptors

C.0 [General]: System architecture—*Data Center, Energy Management, Energy efficiency, Thermal modeling, Optimization, Optimal control, Cyber-physical systems*

## 1. INTRODUCTION

A data center is as a facility used to house computer systems and associated components. As the foundation of the nation's information infrastructure, data centers are growing rapidly in both number and capacity to meet the increasing demands for highly-responsive computing and massive storage. There has been a corresponding growth in the power consumed by data center operations. The Environmental Protection Agency recently reported data centers account for nearly 2.0% of the total energy consumed in the U.S., with a doubling of the power consumed between 2000 and 2006 and a prediction of another doubling in the next five years [25].

With respect to total energy consumption, data center equipment can be broadly classified as information systems (servers, storage devices, routers, etc.) and cooling systems (chillers, air handlers, fans, etc.). As computational density has increased at all levels, from transistors on integrated circuits (ICs), to servers in racks, to racks in a room, the rate at which heat must be removed has increased, leading to nearly equal costs for operating the information systems and cooling systems [2, 5]. Indeed, available cooling capacity has in some cases become the limiting factor on the computational capacity (number of servers per rack) in some data centers [24]. This has motivated considerable research

and development on both fronts: innovations at all levels in low-power storage and computation technology [3]; and major improvements in cooling systems, including better rack and data center architectures to improve cooling system efficiency [17, 16].

In this paper we discuss a unified, coordinated, thermal-computational data center model and we compare the performance of a controller derived using our coupled approach against a controller that disregards the coupling between the computational and thermal aspects of a data center.

The paper is divided as follows: Sec. 2 discusses previous work on energy management for data centers. Section 3 formulates the energy management problem and proposes the unified thermal-computational model. Section 4 discusses a particular instantiation of the data center model, and Sec. 5 presents the simulation results. Finally, Sec. 6 discusses further research directions.

## 2. PREVIOUS WORK

Mechanisms to reduce power consumption in information technology (IT) span dimensional scales from chip level, to room level, and up to the data center level. At the chip level, power state control techniques like dynamic voltage and frequency scaling (DVFS) are largely applied [14, 12]. At the platform level, strategies to reduce the dynamic power consumption of hard disks have been proposed [26, 7]. Researchers have also focused on the network power consumption in data centers [9, 23].

At the room level, power-aware resource management algorithms leverage virtualization [15]. Virtualization allows a server to run different operating systems and applications on the same hardware and also to move the computational workload among servers of a data center. In [6] Lien et al. focus on optimal scheduling algorithm for minimizing server power consumptions. Servers are modeled as M/M/1 queues with only two power states: busy and idle. The results in [6], even though applied to a simplistic data center model, show the capability of power-aware scheduling algorithm to reduce data center power consumption. Some recent papers have considered policies to minimize total data center energy consumption by distributing the workload in a temperature-aware manner [4, 21].

At the data center level, researchers are evaluating techniques to distribute workload among data centers geographically distributed over a wide area, e.g., north America. In [10] the authors discuss under which conditions distributed micro-data centers are less expensive than traditional "mega-data centers". Distributed micro-data centers also of-

fer the possibility to execute the computations at locations where the electricity costs are lower. In [20] the authors consider the implications of electricity price volatility and locational variation with respect to internet scale systems.

Air cooling is the most widely applied solution to control IT temperatures. Liquid cooling is an interesting alternative [18], but costs associated with this approach are preventing its extensive usage. In the room-based air cooling solutions, computer room air conditioner (CRAC) units are associated with the room. At design time, before the actual placement of racks and CRAC units takes place, computational fluid dynamics (CFD) simulations are used to verify that all IT devices will receive the required amount of chilled air.

Row and rack oriented cooling architectures have been proposed [11]. These approaches do not require CFD simulations prior to the rack placement since airflow paths are shorter than those used by the room-oriented solutions and hence, more predictable. Row- and rack-oriented systems are also more efficient than the room-oriented systems, in particular when the power density per rack is higher than 3kW.

Advanced control of cooling systems is discussed by Patel et al. in [16]. A principal innovation of the proposed approach is to integrate the control of the information systems and cooling systems in data centers to achieve levels of energy efficiency that are far lower than the levels that have been achieved thus far using methods that treat these systems separately.

### 3. PROBLEM FORMULATION

The proposed modeling framework is based on a holistic and modular approach that supports: compositional modeling to address alternative data center architectures, multi-scale temporal and spatial abstractions and decompositions, stochastic influences from the environment and workloads and extensibility to incorporate emerging technologies (e.g., solid-state storage, liquid cooling). Based on the level of granularity, thermal and computational nodes, introduced later in this section, can either represent a single device, or an aggregate set of devices, each of which can be modeled by a lower-level thermal and computational network. For example a computational node can represent either a row of server racks, a single blade server in a blade enclosure, or even a single processor located in a blade server.

We model a data center as two coupled networks: a *computational network* and a *thermal network*, as shown in Fig. 1. A data center interacts with the external world executing tasks at the computational network level and consuming electrical power at the thermal network level. Tasks are atomic computation requests.

#### 3.1 Computational network

The computational network describes the evolution of the data center performance (e.g., expected delay per task) over the time. The computational network model is based on the queuing theory, i.e., nodes of the computational network are queuing systems. The computational network is composed of  $n$  nodes, also called computational servers. Tasks are routed in the computational network according to a randomized rule.  $s_{i,j}$ ,  $i, j = 1, \dots, n$  is the probability that a task, after begin executed at a computational node  $i$  is sent to node the computational node  $j$ .  $0 \leq \sum_{j=1}^n s_{i,j} \leq 1$

and  $1 - \sum_{j=1}^n s_{i,j}$  is the probability that a task, after being executed at node  $i$  leaves the data center.

**Computational nodes.** Computational nodes are single server queues with infinite-length buffers. Arrival process to a node  $i$  is given by the combination of the external arrival process and internal scheduling. Tasks are executed at the node  $i$  according to a discipline  $d_i \in \mathcal{D}_i$ , where  $\mathcal{D}_i$  is the discrete, nonempty set of available disciplines at node  $i$ . For example  $\mathcal{D}_i = \{\text{FIFO}, \text{LIFO}, \text{PS}\}^1$ . Each computational server node also has a discrete, nonempty set of available power states  $\mathcal{P}_i$ , i.e.,  $\mathcal{P}_i = \{0, \dots, |\mathcal{P}_i| - 1\}$ , where  $|\mathcal{P}_i|$  represents the number of elements in  $\mathcal{P}_i$ . The expected task execution rate of a computational node is function of the chosen power state. We denote with  $\mu_i(p_i)$  the expected task execution rate at a node  $i$  when the chosen power state is  $p_i$ . If  $p_{i_1} \leq p_{i_2}$  and  $p_{i_1}, p_{i_2} \in \mathcal{P}_i$ , then  $\mu_i(p_{i_1}) \geq \mu_i(p_{i_2})$ . Power state 0 is associated with the highest task execution rate. Finally, we say a node  $i$  is busy when it is executing a task and idle otherwise.

#### 3.2 Thermal network

The thermal network describes the relationship between the overall data center power consumption and the amount of heat exchanged within the data center and with the external environment. The thermal network accounts for the power consumed by CRAC units, IT devices and non-computational devices.

Nodes of the thermal network are divided in three classes: thermal server nodes, CRAC nodes, and environment nodes. Environment nodes represent IT devices other than servers, non-computational devices and external environmental effects (e.g., weather). All nodes in the thermal network have an input temperature  $T_{\text{in}}$ , an output temperature  $T_{\text{out}}$ , and an electrical power consumption  $pw$ . The thermal network is composed of  $n$  thermal server nodes,  $c$  CRAC nodes, and  $e$  environment nodes. Thermal nodes model the *local* relation between power consumption, input temperature and output temperature evolutions. A thermal server node can be used, for example, to describe the evolution of the outlet temperature of a server, given the values of its inlet temperature and power consumption.

We denote with  $T_{\text{in}i}$ ,  $T_{\text{out}i}$ , and  $pw_i$  respectively the input temperature, the output temperature and the power consumption of the thermal node  $i$ .

The input temperature of a thermal node  $i$  is given by a convex linear combination of the output temperatures values of all other nodes, [19]:

$$T_{\text{in}i}(\tau) = \sum_{j=1}^{n+c+e} \gamma_{i,j} T_{\text{out}j}(\tau). \quad (1)$$

Values of  $\gamma_{ij}$  depend upon the particular data center layout.

**Thermal server nodes.** A thermal server node  $j$  is modeled as a first-order linear time-invariant (LTI) system:

$$\dot{T}_{\text{out}j} = k_j(T_{\text{in}j} - T_{\text{out}j}) + c_j \cdot pw_j, \quad (2)$$

where  $k_j$  and  $c_j$  are appropriate positive coefficients. Measurements taken on a server in our laboratory, shown in Fig. 2, validate this model.

**CRAC nodes.** CRAC nodes reduce the input temperatures of thermal nodes. These nodes have an additional in-

<sup>1</sup>FIFO: first in first out, LIFO: last in first out, PS: processor sharing.

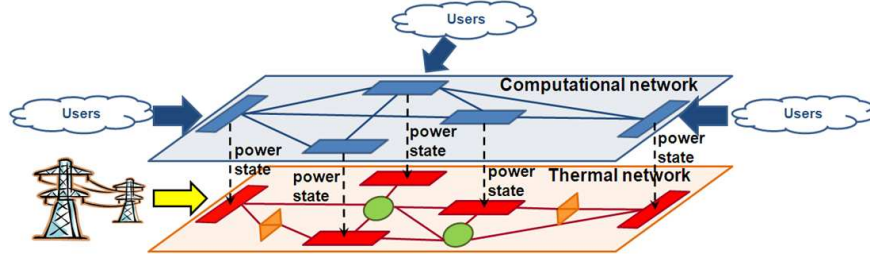


Figure 1: A layered data center model: a *computational network* of server nodes (rectangles); and a *thermal network* of server nodes (rectangles), CRAC nodes (diamonds) and environment nodes (circles). Dashed black arrows denote the particular association law.

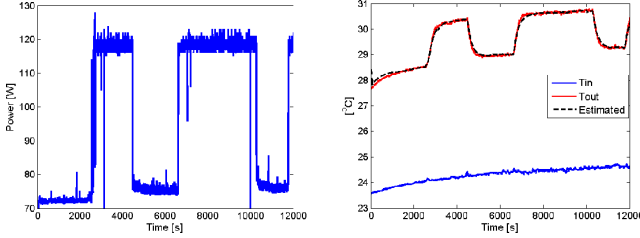


Figure 2: Empirical modeling of thermal server nodes. Left: Server node power consumption. Right: measured  $T_{in}$ ,  $T_{out}$  and predicted  $T_{out}$  from the empirical thermal server node model.

put: the reference output temperature value  $T_{ref}$ . We denote with  $T_{ref_i}$  the reference temperature of the  $i^{th}$  CRAC node. Let  $i$  be a CRAC node, when  $T_{ref_i} \leq T_{in_i}$ ,  $T_{out_i}$  will tend to  $T_{ref_i}$  according to the node dynamic, while  $T_{out_i}$  will tend to  $T_{in_i}$  when  $T_{ref_i} > T_{in_i}$ . The electrical power consumption  $pw_i$  of a CRAC node  $i$  is a function of both  $T_{in_i}$  and  $T_{out_i}$ , monotonically decreasing in  $T_{out_i}$  for all  $T_{out_i} < T_{in_i}$ .

**Environment nodes.** Environment nodes model non-computational devices and non-devices subsystems, such as the external environment, that influence the data center thermal dynamics. For example, the temperature of the environment surrounding the data center can be expressed as the output temperature of an environment node having zero power consumption and output temperature equal to the measured environment temperature.

### 3.3 Integrated thermal-computational model

Each computational server is associated with one thermal server node and we call a *server* the coupling between the two. The  $i^{th}$  server represents then the coupling between the  $i^{th}$  computational node and the  $i^{th}$  thermal node. Power consumption of a server node  $i$  is function of its computational server power. Given the choice of a power state  $p_i \in \mathcal{P}_i$ , the thermal server power consumption is quantized into two values:  $pw_{B,i}(p_i)$  and  $pw_{I,i}(p_i)$ .  $pw_{B,i}(p_i)$  represents the thermal server power consumption when the associated computational server is busy, i.e., the computational server is executing a task.  $pw_{I,i}(p_i)$  represents instead the thermal server power consumption when the associated computational server is idle.

If  $p_{i_1}, p_{i_2} \in \mathcal{P}_i$  and  $\mu_i(p_{i_1}) \geq \mu_i(p_{i_2})$ , then  $pw_{B,i}(p_{i_1}) \geq pw_{B,i}(p_{i_2})$ , i.e., higher task execution rates imply higher

server power consumption.

Based on the level of granularity, thermal and computational nodes can either represent a single device or an aggregate set of devices, each of which can be modeled by a lower-level thermal and computational network. For example a computational node can represent either a row of server racks, a single blade server in a blade enclosure, or even a single processor located in a blade server.

### 3.4 Constraints

The proposed unified thermal-computational model makes it possible to formulate constraints related to both the computational and thermal aspects of a data center. For example, a time delay vs. power consumption curve could be imposed as a joint constraint, rather than imposing independent constraints on each of these variables. Control variables can also be constrained. For example, we could require that no more than  $\bar{n}$  server nodes should be used at the same time, or that two particular computational server nodes should never exchange tasks between each other.

### 3.5 Control Inputs

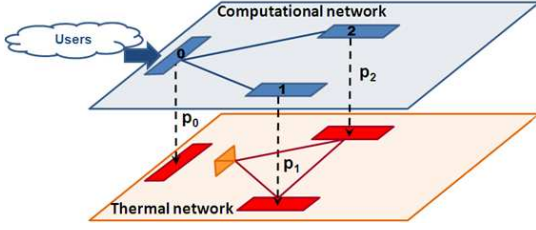
The controllable variables in the proposed data center model are: the scheduling probabilities  $s_{i,j}$ ,  $i, j = 1, \dots, n$ , the thermal server node power states  $p_i$ ,  $i = 1, \dots, n$ , and the CRAC node reference temperatures  $T_{ref_i}$ ,  $i = 1, \dots, c$ . As discussed in [13], standard data center management algorithms, are controllers that try to optimize either the power consumption of server nodes, or the power consumption of the whole data center, relying only on the information contained at the computational network level. Since the information given by the interaction between the thermal and the computational networks are discarded, such an approach leads to sub-optimal solutions.

## 4. EXAMPLE

To illustrate a control strategy derived from the proposed data center model, we consider the data center in Fig. 3. Server nodes are numerated from 0 to 2. Server node 0 will also be called switch.

### 4.1 Computational Network

Tasks arrive at the data center only through the computational switch node. The task arrival process is Poisson distributed with parameter  $\lambda$ . The task sojourn time for the switch node is zero. Tasks are routed by the switch to computational node 1 with probability  $s_{0,1}$  and to computational node 2 with probability  $1 - s_{0,1}$ , i.e., all tasks have



**Figure 3: Data center model.**  $n = 3$  server nodes,  $c = 1$  CRAC nodes,  $e = 0$  environment nodes.

to be routed either to node 1 or to node 2.

When a task is executed at computational server node 1 or 2, it leaves the data center, i.e.,  $s_{1,j} = s_{2,j}$  are constrained to have value 0 for  $j = 0, 1, 2$ .

**Computational server nodes.** Computational server nodes are modeled as M/M/1 queues with infinite length buffers. Their sets of available disciplines are composed only of the FIFO discipline. Computational server node 0 has only one available power state, i.e.,  $\mathcal{P}_0 = \{0\}$ . Server node 1 and 2 have four available power states:  $\mathcal{P}_1 = \mathcal{P}_2 = \{0, 1, 2, 3\}$ . For each admissible power state, computational server node 2 is faster than computational server node 1, i.e.,  $\mu_1(p) \leq \mu_2(p)$  for  $p = 0, 1, 2, 3$ .

## 4.2 Thermal network

Thermal server node 0 does not participate to the thermal energy exchange among other thermal nodes and hence  $\gamma_{0,0} = 1$  and  $\gamma_{0,j} = 0$  for  $j = 1, 2$ . For example this can be due to the thermal isolation of node 0 from the rest of the data center. Since thermal server node 0 does not participate to the thermal energy exchange, we will not consider its temperature evolution. Cooling costs of thermal server node 2 are higher than cooling costs of thermal server node 1.

**Thermal server nodes.** Evolution dynamics of thermal server 1 and 2 are comparable. Thermal server 2 is more energy efficient than thermal server node 1. In particular, for each power state, power consumptions of thermal server node 1 are higher than power consumption of thermal server node 2 either when they are the busy and idle.

**CRAC node.** We denote with  $T_{out3}$ ,  $T_{in3}$  and  $pw_3$  respectively the output temperature, the input temperature and the power consumption of the CRAC node. In this example we assume the dynamic of the CRAC unit is much faster than the dynamic of the thermal servers 1 and 2. When  $T_{ref} > T_{in3}$ , we set  $T_{out3}$  equal to  $T_{in3}$  and the node power consumption is zero. When instead,  $T_{ref} \leq T_{in3}$ , we set  $T_{out3}$  equal to  $T_{ref}$  and the CRAC node power consumption is proportional to  $(T_{in3} - T_{out3})^2$ .

**Environment nodes.** For the sake of a clear exposition in this example we do not consider environment nodes. Absence of environment nodes can happen, for example, when the thermal energy exchange between the data center and the external environment is negligible.

## 4.3 Control approach

We focus on a discrete time control approach and denote the sampling interval with  $\tau$ . Given a choice of controllable variable values that allows the existence of an invariant distribution for the computational network, we assume that

the computational network reaches such distribution in a negligible amount of time.

In order to synthesize a control algorithm for this particular example, we use the theory of stochastic dynamic programming and in particular, the theory of Markov decision process (MDP) [1, 22].

To formulate our optimization problem as a finite MDP we have to identify: a discrete, non-empty set  $\mathcal{X}$  of states, a discrete, finite, non empty set  $\mathcal{A}$  of actions from which the controller can select an action at each time step  $\tau = 0, \dots, k\tau$ ,  $k \in \mathbb{N}$ , a set  $P_{xay}$  of transition probabilities representing the probability of moving from a state  $x$  to a state  $y$  when the action  $a$  is applied, and a function  $c : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}_+$  of immediate costs, where  $\mathbb{R}_+ = \{x \in \mathbb{R} : x \geq 0\}$ .

To reach the goal of a finite set of states and actions we discretize and limit the set of admissible values for  $T_{out1}$ ,  $T_{out2}$ ,  $T_{ref}$ , and  $s_{0,1}$ . Since  $T_{ref}$  and  $s_{0,1}$  are control variables, we can restrict their admissible values to a discrete and finite set. The set of actions is then composed of the tuple  $(T_{ref}, p_1, p_2, s_{0,1})$ , where  $p_1 \in \mathcal{P}_1$  and  $p_2 \in \mathcal{P}_2$ . We consider only action values for which invariant distributions for the computational network exist. This implies that the set of actions is a function of the expected task rate of arrival  $\lambda$  and we denote this set as  $\mathcal{A}(\lambda)$ .

Values of  $T_{out1}$  and  $T_{out2}$  are constrained to evolve according to a difference equation obtained discretizing (2). Hence, their values can span over an infinite set of numbers, even though their input functions assume values over a finite and discrete set. We define  $\tilde{T}_{out1}$  and  $\tilde{T}_{out2}$  as the quantized versions of  $T_{out1}$  and  $T_{out2}$  respectively. The set of states  $\mathcal{X}$  for the MDP will then be given by the tuple  $(\tilde{T}_{out1}, \tilde{T}_{out2})$ . States of the MDP do not include the number of tasks in computational server node 1 and 2 since we assume the computational network reaches the invariant distribution, when it exists, in a negligible amount of time.

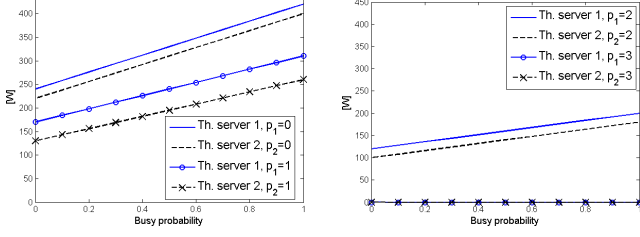
Uncertainty about the true values of  $T_{out1}$  and  $T_{out2}$  is modeled by two independent identical distributed random variables. We assume  $T_{outi}$ ,  $i = 1, 2$  is uniformly distributed in  $[\tilde{T}_{outi,min}(x), \tilde{T}_{outi,max}(x)]$ , where  $\tilde{T}_{outi,min}(x)$  and  $\tilde{T}_{outi,max}(x)$  represent respectively the minimum and the maximum allowed temperature values for  $T_{outi}$ , when its discretized version has value  $\tilde{T}_{outi}(x)$ .  $\tilde{T}_{outi}(x)$  represents the value of  $\tilde{T}_{outi}$  associated with the state  $x \in \mathcal{X}$ . The uncertainty in the true values of  $T_{out1}$  and  $T_{out2}$  and the discrete version of (2) determines each entry of  $P_{xay}$ .

**Constraints.** For any state  $x \in \mathcal{X}$  and  $a \in \mathcal{A}(\lambda)$  we require that the input temperature  $T_{in3}$ ,  $i = 1, 2$  to be always in the interval  $[T_{inmin}, T_{inmax}]$ . In particular we set  $T_{inmin} = 15[^\circ\text{C}]$  and  $T_{inmax} = 31[^\circ\text{C}]$ .

**Immediate costs.** Given a state  $x$  and an action  $a$ , we say that a state  $y \in \mathcal{X}$  is reachable if  $P_{xay} > 0$ . If any reachable state from a state  $x$  does not satisfies the thermal constraint when action  $a$  is chosen, we set the immediate cost of the couple  $(x, a)$  to infinity. If instead, all of the reachable states satisfy the thermal constraint, we set the immediate cost to:

$$c(x, a) = \overline{pw}_1(a) + \overline{pw}_2(a) + \overline{pw}_3(a, x) + f_\alpha(S(a)). \quad (3)$$

In (3)  $\overline{pw}_i(a)$ ,  $i = 1, 2$  represents the expected power consumption of thermal server  $i$  when the action  $a$  is chosen. Let  $i = 1$  and the chosen action be  $\tilde{a} = (\tilde{T}_{ref}, \tilde{p}_1, \tilde{p}_2, \tilde{s}_{0,1}) \in \mathcal{A}(\lambda)$ . Since  $\tilde{a} \in \mathcal{A}(\lambda)$  and due to the assumption of negligible time needed by the computational network to reach the invariant



**Figure 4: Expected power consumption of thermal server node 1 and 2 for different power states.**

distribution, we can compute the busy probability of server 1 as:

$$P_{busy_1} = \frac{\lambda \cdot s_{0,1}}{\mu(\tilde{p}_1)}. \quad (4)$$

The expected power consumption  $\bar{pw}_1(a)$  is then given by:

$$\bar{pw}_1(\tilde{a}) = pw_{B,1}(\tilde{p}_1) \cdot P_{busy_1} + pw_{I,1}(\tilde{p}_1)(1 - P_{busy_1}). \quad (5)$$

$P_{busy_2}$  can be computed similarly.

Figure 4 shows the expected power consumption of thermal server nodes 1 and 2 under different busy probabilities and for all of their admissible power states.

Power consumption of the CRAC node,  $pw_3$ , is null if  $T_{in3} \leq T_{ref}$  and proportional to  $(T_{in3} - T_{ref})^2$  otherwise. Since, we observe the input and output temperature of CRAC node only at discrete points of time and hence, we have to approximate its mean value over the sampling interval time. Let  $\tilde{x} = (T_{out1}, T_{out2}) \in \mathcal{X}$  and  $\tilde{a} \in A$  be respectively the state and the chosen action at time  $\tau$ ,  $\tau = 0, \dots, k\bar{\tau}$ . If at time  $\tau$  the CRAC input temperature  $T_{in3} = \gamma_{3,1}T_{out1} + \gamma_{3,2}T_{out2} + \gamma_{3,3}T_{ref}$  is less than  $T_{ref}$ , then we set  $\bar{pw}_3$  to zero. If at time  $\tau$   $T_{in3}$  is greater than  $T_{ref}$ , then we set  $\bar{pw}_3 \propto (T_{in3} - T_{ref})^2$ . Finally,  $f_\alpha(S(a))$  is monotonic function of the expected sojourn time  $S(a)$ , which is determined by the action  $a \in A(\lambda)$ . The expected sojourn time in the data center can be computed starting from the busy probability of server node 1 and 2, and the value of  $s_{0,1}$ . In this example we set  $f_\alpha(S(a))$  to zero if the expected sojourn time is less than  $\alpha$  and proportional to  $(S(a) - \alpha)^2$  otherwise.

**Cost function.** We choose the following cost function:

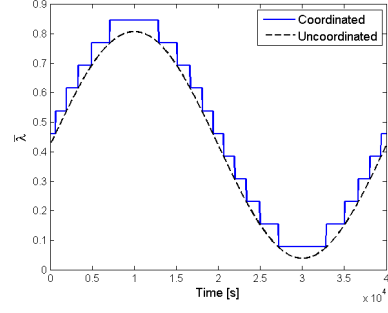
$$J(\mathbf{a}; x(0), \beta) = \mathbb{E} \left[ \sum_{i=0}^{\infty} \beta^i c(x(i), a(i)) \right], \quad (6)$$

where  $\beta \in (0, 1)$ ,  $x(0) \in \mathcal{X}$ ,  $\mathbf{a} = [a(0) a(1) \dots]$ , and  $a(i)$  and  $x(i)$  represent respectively the action and the state at the step  $i$ ,  $a(i) \in A(\lambda)$  and  $x(i) \in \mathcal{X}$  for all  $i = 0, 1, \dots$ . The discount term  $\beta$  ensures the convergence of the cost function to a finite value as  $i$  increases.

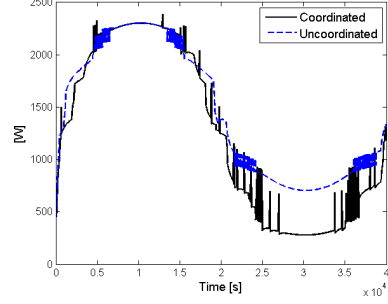
**Control law.** In this example we are interested in stationary policies. A stationary policy is a function  $\pi_\lambda(\cdot)$  from  $\mathcal{X}$  to  $A(\lambda)$ . We denote with  $\Pi_\lambda$  the set of stationary policies from  $\mathcal{X}$  to  $A(\lambda)$ . The control problem is then the research of the optimal stationary policy  $\pi_\lambda^*$  such that:

$$\begin{aligned} \pi_\lambda^* &= \arg \min_{\pi \in \Pi_\lambda} J(\pi, x(0), \beta) = \\ &= \arg \min_{\pi \in \Pi_\lambda} \mathbb{E} \left[ \sum_{i=0}^{\infty} \beta^i c(x(i), \pi(x(i))) \right]. \end{aligned} \quad (7)$$

To solve this control problem, standard approaches like



**Figure 5: Expected task arrival rate.**



**Figure 6: Data center overall power consumption.**

value iteration, policy iteration, or linear programming (LP) can be applied, [1, 22]. We used the Markov Decision Process Toolbox for MATLAB developed by the decision team of the Biometry and Artificial Intelligence Unit of INRA Toulouse [8].

## 5. SIMULATION RESULTS

Performance of the MDP based controller is compared to a controller that tries to minimize the overall data center power consumption disregarding the coupling between the computational and the thermal network. We call *coordinated* the MDP based controller and *uncoordinated* the latter. Let  $A'(\lambda)$  denote the set of tuple  $(p_1, p_2, s_{0,1})$  for which an invariant distribution for the computational network exists.  $s_{0,1}$  can only take values from the same set used for the coordinated controller case. At each time  $\tau = 0, \dots, k\bar{\tau}$ ,  $k \in \mathbb{N}$  the uncoordinated controller chooses an action  $a' \in A'$  in order to minimize the following index:

$$\tilde{J}(a') = \tilde{pw}_1(a') + \tilde{pw}_2(a') + f_\alpha(S(a')), \quad (8)$$

where  $\tilde{pw}_1(a')$ ,  $\tilde{pw}_2(a')$ , and  $f_\alpha(S(a'))$  are computed similarly to the coordinated controller case.

Once an action  $a' \in A'(\lambda)$  is chosen, the uncoordinated controller determines the best value for  $T_{ref}$  in order to enforce the thermal constraints described in Sec. 4 and to minimize the CRAC node power consumption. The choice of the CRAC node reference temperature value is based on the true values of  $T_{out1}$  and  $T_{out2}$  and hence, on the true value of  $T_{in3}$ . The uncoordinated controller has then access to a richer set of thermal information than the coordinated controller.

The uncoordinated controller chooses the best value of  $T_{ref}$  once the other controllable variables have been set. This reflects the behavior of typical data center management al-



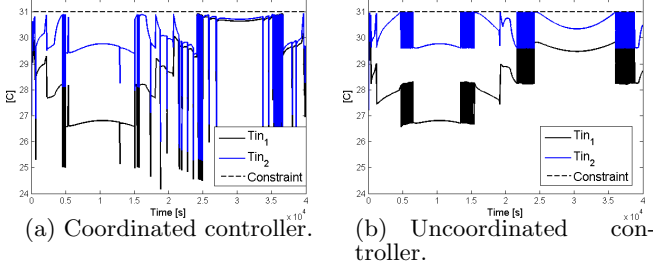


Figure 7: Thermal server input temperature.

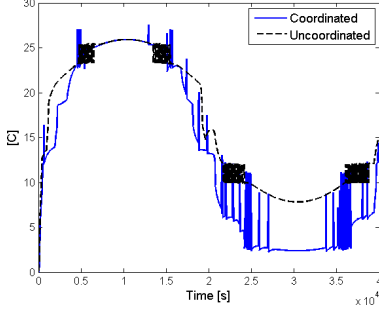


Figure 8: Difference between input and reference temperature of the CRAC node.

gorithms, where the control of the computational and cooling subsystems are performed by independent and uncoordinated controllers. Also, the uncoordinated algorithm can solve its optimization problem in real time, while the coordinated controller requires an off-line computation of a set of policies based on different values of  $\lambda$ .

The coordinated controller then, chooses the best action based on a quantized value of  $\lambda$ . The quantization alphabet corresponds to the values of  $\lambda$  for which an optimal stationary policy was computed.

We compare the coordinated and the uncoordinated controller performance in two different simulations. In both simulations we consider a time varying expected task arrival rate  $\lambda(\tau)$ ,  $\tau = 0, \dots, k\bar{\tau}$ . Values of  $\lambda$  belong to the interval  $[0, \mu_{max})$ , where  $\mu_{max}$  is the data center largest expected task execution rate. In the first simulation we choose a particular time varying function for the task arrival rate  $\lambda$ . In the second simulation instead, we assume that the value of  $\lambda(\tau)$  at time  $\tau$  is the realization of a stochastic process  $\bar{\Lambda}$ .

In the rest of this section we will refer to  $\bar{\lambda} = \frac{\lambda}{\mu_{max}}$  as the normalized expected task arrival rate. Similarly  $\bar{\Lambda}$  will denote the stochastic process from which  $\bar{\lambda}$  takes values.

**Simulation one.** Figure 5 shows the values assumed by  $\bar{\lambda}$  and its quantized version over the time.

Data center overall power consumptions, when using the coordinated and the uncoordinated controller, are shown in Fig. 6. Power consumption of the uncoordinated controller is higher than one provided by the coordinated controller. In particular the difference in the power consumption profiles, is maximized during the low usage periods. This is due to the thermal coupling between computational and thermal networks. During low usage periods, the data center power consumption is dominated by the CRAC node power con-

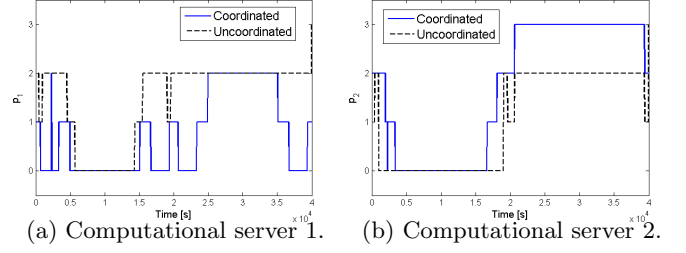


Figure 9: Server power states.

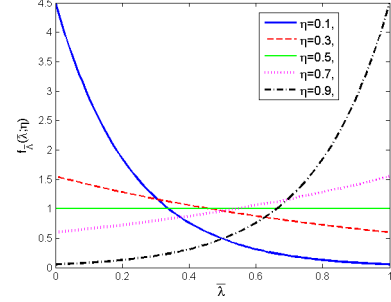


Figure 10: Probability density function of  $\bar{\Lambda}$  for different values of  $\eta$ .

sumption and, as shown in Fig. 8, coordinated controller maintains a smaller input-output temperature difference for the CRAC node, if compared to the uncoordinated one. Figure 7 shows the input temperature of thermal server 1 and 2, dashed line represents the maximum allowed input temperature value. Server power states are shown in Fig. 9.

**Simulation 2.** In this simulation  $\bar{\Lambda}$  is a stochastic process composed of independent identically distributed random variables having probability density function (pdf):

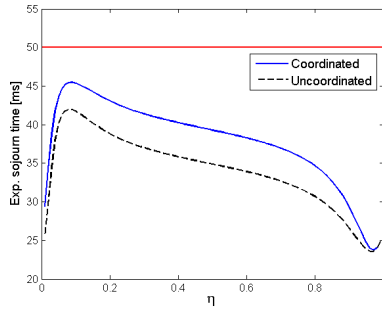
$$f_{\bar{\Lambda}}(\bar{\lambda}; \eta) = \begin{cases} k(\eta) e^{\frac{\eta-0.5}{\eta(1-\eta)} \bar{\lambda}} & \text{if } \bar{\lambda} \in [0, 1) \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

where  $k(\eta)$  is an appropriate coefficient so that  $\int_0^1 f_{\bar{\Lambda}}(x; \eta) dx = 1$  for a given  $\eta \in (0, 1)$ . Figure 10 shows the values of  $f_{\bar{\Lambda}}(\bar{\lambda}; \eta)$  for different values of  $\eta$ . When  $\eta$  tends to 0.5  $f_{\bar{\Lambda}}(\bar{\lambda}; \eta)$  converges to a uniform distribution and we set  $f_{\bar{\Lambda}}(\bar{\lambda}; 0.5) = 1$  for  $\bar{\lambda} \in [0, 1)$  and 0 otherwise. When  $\eta < 0.5$  the expected value of  $\bar{\Lambda}$  is lower than 0.5 and hence, the data center is mostly in a low usage state. When  $\eta > 0.5$  the expected value of  $\bar{\Lambda}$  is higher than 0.5 and hence, the data center is mostly in a high usage state.

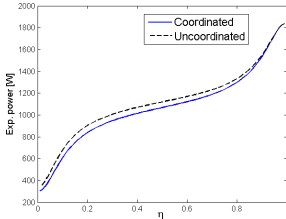
In this simulation we study how the expected sojourn time and the expected overall power consumption vary with respect to the parameter  $\eta$  when values of  $\lambda(\tau)$  are generated from the pdf. in (9). Figure 11 shows that both controllers are able to satisfy the performance constraints for all the admissible values of  $\eta$ . Expected overall data center power consumption is reported in Fig. 12. The coordinated controller outperforms the uncoordinated one for all values of  $\eta$ . The peak power reduction of 13% is obtained for  $\eta \approx 0.04$ .

## 6. DISCUSSION

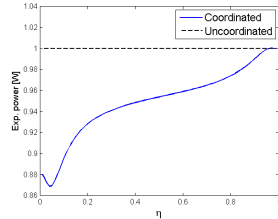
This paper presents a coordinated unified thermal-compu-



**Figure 11: Expected sojourn time of coordinated and uncoordinated controllers.**



(a) Expected power consumption.



(b) Normalized expected power consumption.

**Figure 12: Expected data center power consumption when using the coordinated and the uncoordinated controller.**

tational approach to the modeling and control of data center energy consumption. The proposed model is based on a holistic approach that explicitly represents the relationship between the computational and thermal aspects of a data center. Simulation results from a simple example show that the coordinated controller outperforms the uncoordinated controller in most cases and it is never worse than the uncoordinated one. The development of a unified model able to describe both the computational aspects and the thermal aspects of a data center, is then the key toward a real minimization of the data center overall power consumption.

Current control approach suffers from some issues, first of all the ability to synthesize optimal controllers only for a single and fixed value of  $\lambda$ . We are currently developing a different hierarchical control approach based on model predictive control (MPC). Also our model is composed of multiple parameters that have to be estimated for every specific data center (e.g.,  $\gamma_{i,j}$  coefficients); this requires the development of effective techniques to estimate in real time a large set of parameters.

We are also considering different performance metrics. For example it could be worthwhile to diversify the ratio between task expected sojourn time and overall power consumption, based on the current cost of electricity. We are also studying the usage of service level agreements to minimize complex index that accounts for different user and data center manager requirements.

## 7. ACKNOWLEDGMENTS

This research is supported by PITA under grant C000032167.

## 8. REFERENCES

- [1] E. Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1998.
- [2] C. Bash, C. Patel, A. Shah, and R. Sharma. The sustainable information technology ecosystem. *ITHERM*, May 2008.
- [3] R. Bianchini and R. Rajamony. Power and energy management for server systems. *Computer*, 2004.
- [4] G. F. C. Bash. Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations. Technical report, HPL, August 2007.
- [5] A. J. S. C. D. Patel. Cost model for planning, development and operational of a data center. Technical report, Internet Systems and Storage Laboratory, HP, June 2005.
- [6] C. Lien, Y. Bai, M. Lin, C. Chang, M. Tsai. Web server power estimation, modeling and management. In IEEE, editor, *ICON*, 2006.
- [7] E. V. Carrera, E. Pinheiro, and R. Bianchini. Conserving disk energy in network servers. In *ICS*, 2003.
- [8] I. Chades, M. J. Cros, F. Garcia, R. Sabbadin. Markov decision process (mdp) toolbox v2.0 for matlab. [www.inra.fr/internet/Departements/MIA/T/MDPtoolbox/index.html](http://www.inra.fr/internet/Departements/MIA/T/MDPtoolbox/index.html).
- [9] I. Keslassy, S. T. Chuang, K. Yu, et al. Scaling internet routers using optics. In *SIGCOMM*, Aug. 2003.
- [10] J. H. K. Church, A. Greenberg. On delivering embarrassingly distributed cloud services. In *ACM HotNets*, 2008.
- [11] N. R. K. Dunlap. The advantages of row and rack-oriented cooling architectures for data centers. White paper, APC, 2006.
- [12] K. Govil, E. Chan, H. Wasserman. Comparing algorithm for dynamic speed-setting of a low-power cpu. In *MobiCom*, New York, NY, USA, 1995.
- [13] L. Parolini, B. Sinopoli, B. H. Krogh. Reducing data center energy consumption via coordinated cooling and load management. In *HotPower*, 2008.
- [14] M. Weiser, B. Welch, A. Demers, S. Shenker. Scheduling for reduced cpu energy. In *OSDI*, 1994.
- [15] N. Tolia, Z. Wang, M. Marwah, C. Bash. Delivering energy proportionality with non energy-proportional systems - optimizing the ensemble. *HotPower*, 2008.
- [16] Patel C. D., Bash C. E., Sharma R. K., Beitelmal A., Friedrich R. J. Smart cooling of datacenters. Kauai, HI, July 2003. The PacificRim/ASME International Electronics Packaging Technical Conference and Exhibition.
- [17] Patel, C.D., Sharma, R.K, Bash, C.E., Beitelmal. Thermal considerations in cooling large scale high compute density data centers. San Diego, California, May 2002.
- [18] R. E. Peter Rumsey. Overview of liquid cooling systems. Slides.
- [19] Q. Tang, T. Mukherjee, S. K. S. Gupta, P. Cayton. Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters. Oct. 2006.
- [20] A. Qureshi. Plugging Into Energy Market Diversity. In *7th ACM Workshop on Hot Topics in Networks*

(*HotNets*), Calgary, Canada, October 2008.

- [21] R. Raghavendra, P. Ranganathan, V. Talwar, and Z. Wang. No power struggles: a unified multi-level power management architecture for the data center. In *ASP-LOS*, 2008.
- [22] S. M. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, Jul. 1995.
- [23] S. Nedeveschi, L. Popa, G. Iannaccone, S. Ratnasamy, D. Wetherall. Reducing network energy consumption via sleeping and rate-adaptation. In *NSDI*, 2008.
- [24] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase. Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing*, 9(1):42–49, 2005.
- [25] U.S. Environmental Protection Agency. Report to congress on server and data center energy efficiency. Technical report, ENERGY STAR Program, Aug. 2007.
- [26] A. S. Y. Kim, S. Gurumurthi. Understanding the performance-temperature interactions in disk i/o of server workloads. 2006.