A Hierarchical Approach to Energy Management in Data Centers

Luca Parolini[†], Emanuele Garone[‡], Bruno Sinopoli[†], Bruce H. Krogh[†]

[†]Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213-3890
{lparolin|brunos|krogh}@ece.cmu.edu

[‡]Dipartimento di Elettronica, Informatica e Sistemistica
Universita degli Studi della Calabria
87037 Arcavacata di Rende (CS), Italy
egarone@deis.unical.it

***Abstract*— This paper concerns the management of energy in data centers using a cyber-physical model that supports the coordinated control of both computational and thermal (cooling) resources. On the basis of the structure of the proposed model and practical issues related to the data center layout and distribution of information, we propose a hierarchical optimization scheme in which the higher level chooses goals for regulation at the lower level. Linear programming is applied to solve sequences of one-step look-ahead problems at both the top level and in the lower-level controllers to solve. The approach is illustrated with simulation results.**

I. Introduction

This paper presents a hierarchical control strategy motivated by the problem of energy management in data centers. 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 [8], 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. As computational density has increased at all levels, the rate at which heat must be removed has increased, leading to nearly equal costs for operating the information systems and cooling systems [1], [5].

A good data center workload allocation strategy should consider both the payoff induced by quality of service (QoS) and the cost of powering the servers and computer room air conditioners (CRACs). Higher QoS levels typically lead to higher rates that can be charged to customers. Servers typically have multiple power states with a direct relationship between the power consumed and the QoS offered by the server. Higher power states also lead to increased heat generation. CRAC units must keep the air temperature at the inlets of servers below specified limits to protect the equipment [2], [3], [4]. As the heat that must be removed by CRAC units increases, their average power consumption also increases, leading to higher cooling costs.

The hierarchical control strategy proposed in this paper minimizes the total cost of power consumption minus the pay-off associated to the QoS obtained by each server [4]. In the proposed strategy, local rack controllers manage the amount of resources allocated for executing user requests and local CRAC controllers determine the supplied air temperature of each rack. A central coordinator provides inputs to local controllers in order to guarantee the inlet air temperature does not exceed the specified operating limits. In general, a model predictive control (MPC) approach can be applied at each layer of the hierarchy [6]. A one-step look-ahead controller at both levels of the hierarchy is considered in this paper. This is effective when the transients are fast relative to the optimization period.

The following section presents a model of the computational and thermal dynamics in a data center. Section III discusses the structure and size of this model for a typical data center, providing the motivation for the proposed hierarchical control strategy. Section IV proposes a hierarchical control strategy that uses aggregated information at the higher level that would typically be available for coordinating rack controllers in a data center. The effectiveness of the proposed approach is evaluated through simulation experiments presented in Section V. The concluding section summarizes the contributions of this paper and identifies directions for future research.

This material is based upon work partially supported by the National Science Foundation under Grant No. 0925964. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

II. A Control-Oriented Data Center Model

Let N be the number of servers in a data center, R the number of racks, and C the number of CRAC units. Typically N is a few orders of magnitude larger than C. Let J be the number of different computational services that a data center can provide. During the k^{th} interval, the average arrival rate of user requests in class j is denoted with $\lambda^j(k)$. In the rest of the paper, user requests will be called *jobs*. A scheduler balances the load among different servers so that $\lambda^j(k) = \sum_{i=1}^{N} \lambda_i^j(k)$, where $\lambda_i^j(k)$ represents the average rate of arrival of jobs in class j at the i^{th} server during the k^{th} interval. We consider the case where the scheduling policy is fixed and the scheduler takes a negligible amount of time to route jobs to servers.

Let $\rho_i^j(k)$ represent the average fraction of the total computational resources assigned to jobs in class j by the i^{th} server during the k^{th} interval. For all $i = 1, \ldots, N$, $j = 1, \ldots, C$ and for all $k \in \mathbb{Z}$ $\rho_i^j(k) \in [0, 1]$ and $\sum_{j=1}^{J} \rho_i^j(k) \leq 1$.

Let $\mu_i^j(k)$ be the largest average execution rate of jobs in class j that can be obtained at the i^{th} server during the k^{th} interval. We consider the case where $\mu_i^j(k) = \overline{\mu}_i^j \rho_i^j(k)$,

where $\overline{\mu}_i^j$ is a positive coefficient for all $i = 1, \ldots, N$ and for all $j = 1, \ldots, J$.

Data center payoff depends on service availability and service responsiveness, so we use the average job sojourn time as a measure of QoS.[1] In the rest of the paper we use the term *QoS cost*, the negative of the QoS payoff, rather than *QoS payoff*, since we formulate the optimization problem in terms of minimizing cost rather than maximizing profit.

The proposed control approach approximates the average job sojourn time with the difference, at every time k, between $\mu_i^j(k)$ and $\lambda_i^j(k)$. The idea behind the proposed approximation stems from the analysis of the expected sojourn time in the M/M/1 queuing systems: when the expected service rate μ is larger than the expected arrival rate λ, then the expected sojourn time is given by $(\mu - \lambda)^{-1}$ and it equals the long-run average sojourn time of jobs in the queue. In such a case, minimizing the average sojourn time is equivalent to maximize the difference between μ and λ.

Let $c_{q,i}^j(k)$ denote the QoS cost obtained at the i^{th} server during the k^{th} interval for jobs in class j, that is,

$$c_{q,i}^j(k) = \mathsf{c}_{q,i}^j(\lambda_i^j(k) - \overline{\mu}_i^j \rho_i^j(k)), \tag{1}$$

where $\mathsf{c}_{q,i}^j$ is a non-negative constant. The total data center QoS cost is defined as

$$c_q(k) = \sum_{i=1}^{N}\sum_{j=1}^{J} c_{q,i}^j(k) = \mathbf{c}_q^T(\boldsymbol{\lambda}(k) - diag\{\overline{\boldsymbol{\mu}}\}\boldsymbol{\rho}(k)), \tag{2}$$

where $\mathbf{c}_q^T$ is the vector that collects the $\mathsf{c}_{q,i}^j$ coefficients, $\boldsymbol{\lambda}(k) = [\lambda_1^1(k), \ldots, \lambda_1^J(k), \lambda_2^1(k), \ldots, \lambda_N^J(k)]^T$, $\overline{\boldsymbol{\mu}} = [\overline{\mu}_1^1, \ldots, \overline{\mu}_1^J, \overline{\mu}_2^1, \ldots, \overline{\mu}_N^J]^T$, $diag\{\overline{\boldsymbol{\mu}}\}$ is the diagonal matrix obtained by placing the elements of the vector $\overline{\boldsymbol{\mu}}$ along the main diagonal, and $\boldsymbol{\rho}(k) = [\rho_1^1(k), \ldots, \rho_1^J(k), \rho_2^1(k), \ldots, \rho_N^J(k)]^T$.

Servers increase their power consumption as the amount of computational resources used to execute jobs increases. We assume the total power consumed by a the i^{th} server is given by the following quadratic relationship between the power consumed by each job type and the arrival rate for each job type

$$\mathsf{p}_i(k) = \boldsymbol{\lambda}_i^T(k) C_{\mathsf{p},i} \boldsymbol{\rho}_i(k), \tag{3}$$

where $\boldsymbol{\rho_i}(k) = [\rho_i^1(k), \ldots, \rho_i^J(k)]^T$, $\boldsymbol{\lambda}_i(k) = [\lambda_i^1(k), \ldots, \lambda_i^J(k)]^T$, and $C_{\mathsf{p},i}$ is a $J \times J$ positive-definite matrix.

We order the inlet and outlet temperatures of servers and CRAC units using indices 1 to N for the inlet and the outlet temperatures of serves, and indices $N+1$ to $N+C$ for the inlet and the outlet temperatures of CRAC units. Let $T_{\text{in},i}(k)$ and $T_{\text{out},i}(k)$ represent respectively the inlet and the outlet air temperature of the i^{th} server at the beginning of the k^{th} interval. As discussed in [3], [4], the evolution of $T_{\text{out},i}(k)$ can be modeled as

$$\begin{aligned} T_{\text{out},i}(k+1) = (1 - \mathsf{k}_i) T_{\text{out},i}(k) + \mathsf{k}_i T_{\text{in},i}(k) + \\ \mathsf{c}_{\mathsf{p},i} \mathsf{p}_i(k), \end{aligned} \tag{4}$$

[1]The job sojourn time of is defined as the difference between the time when a job arrives at a server and the time when it leaves the data center.

where k_i is the (discrete-time) thermal coefficient of the i^{th} server, $\mathsf{c}_{\mathsf{p},i}$ is a non-negative coefficient, and $\mathsf{p}_i(k)$ is the average power consumption of the server during the k^{th} interval.

We consider the case where CRAC units have a colocated controller. The input to the controller of the i^{th} CRAC is the reference temperature of the i^{th} CRAC, $T_{\text{ref,i}}(k)$. Since a CRAC unit can only be used to cool the air, we assume the colocated controller will make $T_{\text{out},i}(k)$ tend to the reference temperature only when $T_{\text{out},i}(k)$ is smaller than the CRAC inlet air temperature. For CRAC units, the supplied air temperature evolution is modeled as

$$\begin{aligned} T_{\text{out},i+N}(k+1) = (1 - \mathsf{k}_{i+N}) T_{\text{out},i+N}(k) + \\ \mathsf{k}_{i+N} \min\{T_{\text{ref,i}}(k), T_{\text{in},i+N}(k)\}, \end{aligned} \tag{5}$$

which can be rewritten in terms of linear constraints as

$$\begin{aligned} T_{\text{out},i+N}(k+1) = (1 - \mathsf{k}_{i+N}) T_{\text{out},i+N}(k) + \\ \mathsf{k}_{i+N} T_{\text{in},i+N}(k) + \mathsf{k}_{i+N} \Delta T_{\text{ref,i}}(k) \\ \Delta T_{\text{ref,i}}(k) \leq 0, \end{aligned} \tag{6}$$

where $\Delta T_{\text{ref,i}}(k) = T_{\text{ref,i}}(k) - T_{\text{in},i+N}(k)$ is a fictitious reference signal that requires the colocated CRAC controller to keep the supplied air temperature $\Delta T_{\text{ref,i}}(k)$ below the CRAC inlet air temperature.

As discussed in [7], the inlet air temperature of servers and CRAC units can be approximated by a linear combination of the output temperatures of all other servers and the air temperatures supplied by the CRAC units

$$T_{\text{in},i}(k) = \sum_{j=1}^{N} \gamma_{i,j} T_{\text{out},j}(k) + \sum_{j=1}^{C} \gamma_{i,j+N} T_{\text{out},j+N}(k), \tag{7}$$

where $\gamma_{i,j}$ is the coefficient that relates $T_{\text{out},j}(k)$ to $T_{\text{in},i}(k)$, and $T_{\text{out},j+N}(k)$ is the supplied air temperature of the j^{th} CRAC at time k.

Define $\mathcal{C} = \{N+1, \ldots, N+C\}$ as the set of indexes of CRAC units and let $\mathcal{R}_i$ be the set of indexes of servers located in the i^{th} rack, $\boldsymbol{T}_{\text{out},[\mathcal{R}_i]}(k)$ be the vector of the outlet temperatures of the servers in the i^{th} rack, and n_i be the number of servers in the i^{th} rack. Constraints on the server inlet air temperature are given as

$$\boldsymbol{T}_{\text{in},[\mathcal{R}_i]}(k) \leq \overline{\boldsymbol{T}_{\text{in},[\mathcal{R}_i]}}, \qquad i = 1, \ldots, R. \tag{8}$$

The $\gamma_{i,j}$ coefficients can be collected in the matrix Γ defined as $[\Gamma]_{i,j} = \gamma_{i,j}$ for all $i, j = 1, \ldots, N+C$. These parameters need to be determined empirically, but this is usually not possible because temperatures are generally not measured in enough places in a data center to provide sufficient data to construct the complete, detailed model. It is also the case that the thermal interaction of servers located far apart from each other is negligible, i.e., $\gamma_{i,j}$ is negligibly small when i, j correspond to locations significantly distant from each other. Therefore, it is common practice in data center applications, to consider a reduced order version of (8), where the model parameters are defined only for locations where a temperature sensor have been placed. For the reduced-order model, let $\boldsymbol{T}_{\text{out},\mathcal{R}_i}(k)$ and $\boldsymbol{T}_{\text{in},\mathcal{R}_i}(k)$ be, respectively, the vector of the outlet and the inlet temperatures

collected at the i^{th} rack at the beginning of the k^{th} interval. For example, the two vectors can represent the measured temperature at the bottom, middle, and top level of the rack. For every rack, the value of its inlet and outlet temperature vector is given by a convex combination of the inlet and outlet temperatures of a server

$$\boldsymbol{T}_{\text{in},\mathcal{R}_i}(k) = G_{in,i}\boldsymbol{T}_{\text{in},[\mathcal{R}_i]}(k), \tag{9}$$

$$\boldsymbol{T}_{\text{out},\mathcal{R}_i}(k) = G_{out,i}\boldsymbol{T}_{\text{out},[\mathcal{R}_i]}(k), \tag{10}$$

where the values of the matrices $G_{in,i}$ and $G_{out,i}$ depend on the position of the temperature sensors and on the server air flows. We assume that $G_{in,i}$ and $G_{out,i}$ are full row rank matrices.

Let v be the rack index where the i^{th} server is located. Eq. (7) can be approximated as

$$\begin{aligned} T_{\text{in},i}(k) = &\sum_{j\in\mathcal{R}_v}\gamma_{i,j}T_{\text{out},j}(k) + \sum_{j\in\mathcal{C}}\gamma_{i,j}T_{\text{out},j}(k) + \\ &\sum_{\substack{j=1\\ j\neq v}}^{R}\boldsymbol{\gamma}_{i,\mathcal{R}_j}\boldsymbol{T}_{\text{out},\mathcal{R}_j}(k), \end{aligned} \tag{11}$$

where $\boldsymbol{\gamma}_{i,\mathcal{R}_j}$ is the vector which represents the relative global effect of the j^{th} rack on the i^{th} server.

We consider each rack and each CRAC unit as a different subsystem of the data center and let $\boldsymbol{x}_i(k)$ be the state of the i^{th} subsystem. In particular, for $i = 1,\ldots,R$ (racks) $\boldsymbol{x}_i(k) = T_{\text{out},[\mathcal{R}_i]}(k)$ and $\boldsymbol{z}_i(k) = \boldsymbol{T}_{\text{out},\mathcal{R}_i}(k)$, while for $i = R+1,\ldots,R+C$ (CRACs), $\boldsymbol{x}_i(k) = T_{\text{out},i}(k)$ and $\boldsymbol{z}_i(k) = T_{\text{out},i}(k)$. The overall state of the data center is then

$$\boldsymbol{x}(k) = \left[\,\boldsymbol{x}_1^T(k),\ldots,\boldsymbol{x}_{R+C}^T(k)\,\right]^T \tag{12}$$

and the dynamics of both servers and CRAC units can be written as

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= A_i\boldsymbol{x}_i(k) + B_i(k)\boldsymbol{u}_i(k) + \sum_{\substack{j=1\\ j\neq i}}^{R+C} B_{i,j}\boldsymbol{z}_j(k) \\ \boldsymbol{z}_i(k) &= G_i\boldsymbol{x}_i(k), \end{aligned} \tag{13}$$

where for $i = 1,\ldots,R$

$$A_i = I - diag\{\mathsf{k}_{[\mathcal{R}_i]}\} + diag\{\mathsf{k}_{[\mathcal{R}_i]}\}\Gamma_{[\mathcal{R}_i,\mathcal{R}_i]}, \tag{14}$$

$$B_i(k) = diag_B\{\mathsf{c}_{\mathsf{p},i}\boldsymbol{\lambda}_i^T(k)C_{\mathsf{p},i}\}, \tag{15}$$

$$B_{i,j} = diag\{\mathsf{k}_{[\mathcal{R}_i]}\}\Gamma_{[\mathcal{R}_i],j}, \tag{16}$$

$$\boldsymbol{u}_i(k) = [\boldsymbol{\rho}_{i_1}^T,\ldots,\boldsymbol{\rho}_{i_{n_i}}^T]^T, \tag{17}$$

$$G_i = G_{out,i}. \tag{18}$$

In the above equations, the operator $diag_B\{X_i\}$ converts the sequence of $\{X_i\}$ matrices in the block diagonal matrix X such that its i^{th} diagonal block is X_i. For $i = R+1,\ldots,R+C$, we have

$$A_i = 1 - \mathsf{k}_i(1-\gamma_{i,i}), \tag{19}$$

$$B_i(k) = \mathsf{k}_i, \tag{20}$$

$$B_{i,j} = \mathsf{k}_i diag\{\boldsymbol{\gamma}_{i,j}\}, \tag{21}$$

$$\boldsymbol{u}_i(k) = \Delta T_{\text{ref,i}}(k), \tag{22}$$

$$G_i = 1, \tag{23}$$

where the vector $\boldsymbol{\gamma}_{i,j}$ represents the effect of the j^{th} rack on the i^{th} CRAC unit for $j = 1,\ldots,R$, while it represents the effect of the $(R-j)^{th}$ CRAC unit on the i^{th} one for $j = R+1,\ldots,R+C$. Therefore, we can write $\boldsymbol{\gamma}_{i,j} = \boldsymbol{\gamma}_{i,\mathcal{R}_j}$ for $j = 1,\ldots,R$ and $\boldsymbol{\gamma}_{i,j} = \gamma_{i,N-R+j}$ for $j = R+1,\ldots,R+C$.

Let $\mathsf{c}_e(k)$ represent the average electricity cost over the k interval. The total server electricity cost is given by

$$c_{\mathsf{p}}(k) = \mathsf{c}_e(k)\sum_{i=1}^{N}\mathsf{p}_i(k) = \boldsymbol{e}(k)^T\boldsymbol{\rho}(k), \tag{24}$$

where c_e is the electricity cost.

CRAC unit power consumption is in general a nonlinear function of CRAC inlet and outlet temperatures reflecting the fact that power efficiency increases as the outlet air temperature [2]. In order to force the CRAC units to keep their outlet temperature at the highest value that does not violate the temperature constraints of servers, we consider the following cost

$$c_{T_{\text{ref}}}(k) = \mathbf{c}_{T_{\text{ref}}}^T\Delta\boldsymbol{T}_{\text{ref}}, \tag{25}$$

where $\mathbf{c}_{T_{\text{ref}}}^T \leq 0$.

The total data center operating cost can now be expressed as

$$\begin{aligned} &c_q(k) + c_{\mathsf{p}}(k) + c_{T_{\text{ref}}}(k) = \\ &\quad \mathbf{c}_q^T\boldsymbol{\lambda}(k) - \boldsymbol{c}_q^T diag\{\overline{\boldsymbol{\mu}}\}\boldsymbol{\rho}(k) + \\ &\quad \boldsymbol{e}^T(k)\boldsymbol{\rho}(k) + \boldsymbol{c}_{T_{\text{ref}}}^T(k)\Delta\boldsymbol{T}_{\text{ref}}(k). \end{aligned} \tag{26}$$

Constraints on each of the $\boldsymbol{u}_i(k)$ variables can be written as

$$0 \leq \boldsymbol{u}_i(k) \leq 1, \quad i = 1,\ldots,R, \tag{27}$$

$$\mathbf{1}^T\boldsymbol{u}_i(k) \leq 1, \quad i = 1,\ldots,R, \tag{28}$$

$$\boldsymbol{u}_i(k) \leq 0, \quad i = R+1,\ldots,R+C. \tag{29}$$

For all $i = 1,\ldots,R$, constraints on the vector of the rack inlet air temperatures can be written as

$$G_i\Gamma_{[\mathcal{R}_i],[\mathcal{R}_i]}\boldsymbol{x}_i(k) + G_i\sum_{\substack{j=1\\ j\neq i}}^{R+C}B_{i,j}\boldsymbol{z}_j(k) \leq \overline{\boldsymbol{T}_{\text{in},\mathcal{R}_i}}. \tag{30}$$

III. Control of a data center: The Dimensionality Challenge

As seen in Section II, a data center can be modeled as a linear system composed of S linear time-varying subsystems of the form

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= A_i\boldsymbol{x}_i(k) + B_i(k)\boldsymbol{u}_i(k) + B_{i,z}\boldsymbol{z}(k) \\ \boldsymbol{z}_i(k) &= G_i\boldsymbol{x}_i(k), \end{aligned} \tag{31}$$

where $\boldsymbol{x}_i(k) \in \mathbb{R}^{n_i}$, $\boldsymbol{u}_i(k) \in \mathbb{R}^{p_i}$, and $\boldsymbol{z}_i(k) \in \mathbb{R}^{m_i}$ are respectively the state, the input, and the output of the i^{th} subsystem. In particular, for the data center model described in the previous section, we have $m_i < n_i$ and $p_i = n_i J$ for $i = 1,\ldots,R$ (racks). The dimension of the output of each rack-related subsystem is much smaller than the dimension

of the input. The matrix $B_{i,z}$ accounts for all of the $B_{i,j}$ matrices and $B_{i,i}$ is zero for all $i = 1, \ldots, S$. The vector $\boldsymbol{z}(k) = \left[\, \boldsymbol{z}_1^T(k), \ldots, \boldsymbol{z}_S^T(k) \,\right]^T$ is the vector of all subsystem outputs.

The system in (31) is subject to the following input and output linear constraints

$$\begin{aligned} \underline{\boldsymbol{u}}_i \leq \quad & \boldsymbol{u}_i(k) && \leq \overline{\boldsymbol{u}}_i && (32)\\ & H_i \boldsymbol{u}_i(k) && \leq \mathbf{1} && (33)\\ & F_i \boldsymbol{x}_i(k) + F_{i,z} \boldsymbol{z}(k) && \leq \overline{\boldsymbol{z}}_i, && (34) \end{aligned}$$

where $F_i \in \mathbb{R}^{m_{z,i} \times n_i}$, $m_{z,i} < n_i$ and $F_{i,z} \in \mathbb{R}^{m_{z,i} \times m_i}$.

Since we focus on the optimization of a discrete-time linear system subject to constraints on the input, model predictive control (MPC) is a natural control approach. Assuming that the state of the overall system at time k enforces (34), at each time step we want to solve the following optimization problem

$$\begin{aligned} &\min_{\mathcal{U}_{1,\tau}} \sum_{i=1}^{S} \sum_{j=0}^{\tau-1} \boldsymbol{c}_{i,u}^T(k) \hat{\boldsymbol{u}}_i(k+j|k) \\ &\textbf{s.t.} \\ &\underline{\boldsymbol{u}}_i \leq \hat{\boldsymbol{u}}_i(k+j|k) \leq \overline{\boldsymbol{u}}_i, \; j = 0, ..., \tau-1, i = 1, ...S \\ &H_i \hat{\boldsymbol{u}}_i(k+j|k) \leq 1, \qquad j = 0, ..., \tau-1, i = 1, ...S \\ &F_i \hat{\boldsymbol{x}}_i(k+j|k) + F_{i,z} \hat{\boldsymbol{z}}(k+j|k) \leq \overline{\boldsymbol{z}}_i, \\ &\qquad\qquad\qquad\qquad j = 1, ..., \tau, i = 1, ..., S \\ &\mathcal{U}_{1,\tau} = \left\{ \hat{\boldsymbol{u}}_1(k|k), \ldots, \hat{\boldsymbol{u}}_S(k+\tau-1|k) \right\} \end{aligned} \tag{35}$$

where the predicted variables are

$$\begin{aligned} &\hat{\boldsymbol{x}}_i(k+j+1|k) = A_i \hat{\boldsymbol{x}}_i(k+j|k) + \\ &\qquad B_i(k+j) \hat{\boldsymbol{u}}_i(k+j|k) + B_{i,z} \hat{\boldsymbol{z}}(k+j|k) \\ &\hat{\boldsymbol{z}}_i(k+j|k) = G_i \hat{\boldsymbol{x}}_i(k+j|k) \end{aligned} \tag{36}$$

and it is assumed $\hat{\boldsymbol{x}}_i(k|k) = \boldsymbol{x}_i(k)$.

Data centers are large-scale systems. In such a scenario it could be too complex to collect data from all the sensors and compute all the control actions with a single controller that closes the loop at each sampling step. In the proposed hierarchical strategy: (i) a central coordinator specifies the external behavior $\boldsymbol{z}(k)$ of each subsystems minimizing a global cost function; (ii) local regulators, one for each subsystem, optimize the local cost functions with the additional constraint of ensuring that their own external behavior $\boldsymbol{z}_i(k)$ is coherent with the value specified by the coordinator.

IV. A Hierarchical Control Strategy

This section discusses the optimization problem in (35) for the case $\tau = 1$. Although it may seem restrictive, optimizing only over the one step ahead prediction can be an appropriate solution when the predictive values of future job arrival rate have a large variability and hence the predictive cost values have weak relevance compared to the current estimated one.

The one-step optimization problem is given by

$$\begin{aligned} &\min_{\mathcal{U}_1} \sum_{i=1}^{S} \boldsymbol{c}_{i,u}^T(k) \hat{\boldsymbol{u}}_i(k|k) \\ &\textbf{s.t.} \\ &\underline{\boldsymbol{u}}_i \leq \hat{\boldsymbol{u}}_i(k|k) \leq \overline{\boldsymbol{u}}_i, \qquad i = 1, ...S \\ &H_i \hat{\boldsymbol{u}}_i(k|k) \leq \mathbf{1} \\ &F_i B_i(k) \hat{\boldsymbol{u}}_i(k|k) + \sum_{\substack{j=1 \\ j \neq i}}^{S} F_{i,z_j} G_j B_j(k) \hat{\boldsymbol{u}}_j(k|k) + \mathbf{k}_i(k) \leq \overline{\boldsymbol{z}}_i, \\ &\qquad i = 1, ..., S \\ &\mathcal{U}_1 = \left\{ \hat{\boldsymbol{u}}_1(k|k), \ldots, \hat{\boldsymbol{u}}_S(k|k) \right\}, \end{aligned} \tag{37}$$

where F_{i,z_j} is the part of matrix $F_{i,z}$ related to the sub-vector $\hat{\boldsymbol{u}}_j(k|k)$ and the vector $\mathbf{k}_i(k)$ is given by

$$\begin{aligned} \mathbf{k}_i(k) = F_i A_i \boldsymbol{x}_i(k) + F_i B_{i,z} \boldsymbol{z}(k) + \\ \sum_{\substack{j=1 \\ j \neq i}}^{S} F_{i,z_j} \Big(G_j A_j \boldsymbol{x}_j(k) + \sum_{\substack{h=1 \\ h \neq j}}^{S} G_j B_{j,h} G_h \boldsymbol{x}_h(k) \Big). \end{aligned}$$

The relevant feature of the optimization problem in (37) is that the only part of the i^{th} subsystem which affects all of the other subsystems is $G_i B_i \hat{\boldsymbol{u}}_i(k|k)$. Therefore, the contribution of the i^{th} subsystem to the evolution of all of the other subsystems lives in a space of dimension m_i which is much smaller than the dimension of the i^{th} system input.

Assume that for every i and k, $G_i B_i(k)$ is a full row rank matrix. For each of the i subsystems we define a $p_i \times m_i$ matrix $M_i(k)$ such that $G_i B_i(k) M_i(k) = I$. We can now consider a new two-stage optimization, where at first, the optimization is performed over a set of much smaller dimension and then each local regulator solves its own optimization problem with the additional constraint that its local action has to lead to the same output chosen at the upper level of the hierarchy.

The first part of the two-stage optimization problem is

$$\begin{aligned} &\min_{\mathcal{V}_1} \sum_{i=1}^{S} \boldsymbol{c}_{i,u}^T(k) M_i(k) \hat{\boldsymbol{v}}_i(k|k) \\ &\textbf{s.t.} \\ &\underline{\boldsymbol{u}}_i \leq M_i \hat{\boldsymbol{v}}_i(k|k) \leq \overline{\boldsymbol{u}}_i, \qquad i = 1, ...S \\ &H_i M_i(k) \hat{\boldsymbol{v}}_i(k|k) \leq 1 \qquad i = 1, ...S \\ &F_i B_i(k) M_i(k) \hat{\boldsymbol{v}}_i(k|k) + F_{i,z} \hat{\boldsymbol{v}}(k|k) + \mathbf{k}_i(k) \leq \overline{\boldsymbol{z}}_i. \\ &\qquad\qquad\qquad\qquad\qquad\qquad i = 1, ..., S, \\ &\mathcal{V}_1 = \{ \hat{\boldsymbol{v}}_1(k|k), \ldots, \hat{\boldsymbol{v}}_S(k|k) \} \end{aligned} \tag{38}$$

where $\hat{\boldsymbol{v}}^*(k|k) = [\hat{\boldsymbol{v}}_1^{*T}(k|k), \ldots, \hat{\boldsymbol{v}}_S^{*T}(k|k)]^T$. The vector $\hat{\boldsymbol{v}}^*(k|k)$, solution of (38), is then broadcasted to each of the i^{th} subsystems, which solve problems of the following form:

$$\begin{aligned} &\min_{\hat{\boldsymbol{u}}_i(k|k)} \boldsymbol{c}_{i,u}^T(k) \hat{\boldsymbol{u}}_i(k|k) \\ &\textbf{s.t.} \\ &\underline{\boldsymbol{u}}_i \leq \hat{\boldsymbol{u}}_i(k|k) \leq \overline{\boldsymbol{u}}_i, \\ &H_i \hat{\boldsymbol{u}}_i(k|k) \leq \mathbf{1} \\ &F_i B_i(k) \hat{\boldsymbol{u}}_i(k|k) + \sum_{\substack{l=1 \\ l \neq i}}^{S} F_{i,z_l} \hat{\boldsymbol{v}}_l(k|k) + \mathbf{k}_i(k) \leq \overline{\boldsymbol{z}}_i, \\ &\hat{\boldsymbol{v}}_i^*(k|k) = G_i B_i(k) \hat{\boldsymbol{u}}_i(k|k), \end{aligned} \tag{39}$$

where the last constraint ensures the coherence of the optimization of each subsystem.

Proposition 1: The following properties hold true: (i) the minimum cost of (38) is always greater than or equal to the minimum cost of (37); (ii) if the optimization problem (38) is feasible, then (39) is feasible for all $i = 1, \dots, S$; (iii) the minimum cost of the i^{th} sub-problem in (39) is smaller than or equal to $\boldsymbol{c}_{i,u}^T(k) M_i(k) \hat{\boldsymbol{v}}_i^\star(k|k)$, where $\hat{\boldsymbol{v}}_i^\star(k|k)$ is the i^{th} sub-vector of the solution to (38); (iv) if (39) is feasible for every i, then the sum of the minimum costs of the optimization problem of each subsystem is grater than or equal to the minimum cost of (37).

Proof: Follows by construction. ∎

Proposition 2: The condition

$$\hat{\boldsymbol{v}}_i^\star(k|k) = B_i(k) M_i(k) \hat{\boldsymbol{u}}_i(k|k)$$

in (39) can be replaced by

$$F_{j,z_i} M_i(k) \hat{\boldsymbol{u}}_i(k|k) \leq F_{j,z_i} \hat{\boldsymbol{v}}_i^\star(k|k) \qquad (40)$$

$$\text{for all } i, j = 1, \dots, S,\ i \neq j.$$

Proof: Let $\hat{\boldsymbol{v}}(k|k)^\star$ be a solution of (38) and consider $\tilde{\boldsymbol{u}}(k|k) = [\tilde{\boldsymbol{u}}_1^T(k|k), \dots, \tilde{\boldsymbol{u}}_S^T(k|k)]^T$ such that each sub-vector $\tilde{\boldsymbol{u}}_i^T(k|k)$ belongs to the feasible set of

$$\begin{aligned}
&\min_{\hat{\boldsymbol{u}}_i(k|k)} \ \boldsymbol{c}_{i,u}^T(k) \hat{\boldsymbol{u}}_i(k|k) \\
&\textbf{s.t.} \\
&\underline{\boldsymbol{u}}_i \leq \hat{\boldsymbol{u}}_i(k|k) \leq \overline{\boldsymbol{u}}_i, \\
&H_i \hat{\boldsymbol{u}}_i(k|k) \leq \mathbf{1} \\
&F_i B_i(k) \hat{\boldsymbol{u}}_i(k|k) + \textstyle\sum_{\substack{j=1 \\ j \neq i}}^S F_{i,z_j} \hat{\boldsymbol{v}}_j^\star(k|k) + \mathbf{k}_i(k) \leq \overline{\boldsymbol{z}}_i, \\
&F_{j,z_i} M_i(k) \hat{\boldsymbol{u}}_i(k|k) \leq F_{j,z_i} \hat{\boldsymbol{v}}_i^*(k|k) \\
&\qquad\qquad \text{for all } i, j = 1, \dots, S,\ i \neq j.
\end{aligned} \qquad (41)$$

Therefore, for all $i = 1, \dots, S$

$$F_i B_i(k) \tilde{\boldsymbol{u}}_i(k|k) + \sum_{\substack{j=1 \\ j \neq i}}^S F_{i,z_j} G_j B_j \tilde{\boldsymbol{u}}_j^\star(k|k) + \mathbf{k}_i(k) \leq \overline{\boldsymbol{z}}_i. \qquad (42)$$

This implies that the vector $\tilde{\boldsymbol{u}}(k|k)$ is a feasible point for (37).

We now prove that the feasible set of (39) is contained in the feasible set of (41). Let $\bar{\boldsymbol{u}}(k|k) = [\bar{\boldsymbol{u}}_1^T(k|k), \dots, \bar{\boldsymbol{u}}_S^T(k|k)]^T$ such that each sub-vector $\bar{\boldsymbol{u}}_i^T(k|k)$ is a feasible point for the i^{th} problem in (39). Then we have

$$\begin{aligned}
&\underline{\boldsymbol{u}}_i \leq \hat{\boldsymbol{u}}_i(k|k) \leq \overline{\boldsymbol{u}}_i \\
&H_i \hat{\boldsymbol{u}}_i(k|k) \leq \mathbf{1} \\
&F_i B_i(k) \hat{\boldsymbol{u}}_i(k|k) + \sum_{\substack{j=1 \\ j \neq i}}^S F_{i,z_j} G_j B_j(k) \hat{\boldsymbol{u}}_j(k|k) \leq \overline{\boldsymbol{z}}_i - \mathbf{k}_i(k) \\
&\hat{\boldsymbol{v}}_i^*(k|k) = G_i B_i(k) \hat{\boldsymbol{u}}_i(k|k)
\end{aligned}$$

and hence, every $\bar{\boldsymbol{u}}_i^T(k|k)$ is a feasible point for (41). ∎

Proposition 3: Let $\tilde{\boldsymbol{u}}_i(k|k)$ and $\hat{\boldsymbol{v}}_i(k|k)$ be such that $G_i B_i(k) \tilde{\boldsymbol{u}}_i(k|k) = \hat{\boldsymbol{v}}_i(k|k)$, then there exists a vector $\boldsymbol{\xi}_i \in \mathcal{N}(G_i B_i(k))$ such that $\tilde{\boldsymbol{u}}_i(k|k) = M_i(k) \hat{\boldsymbol{v}}_i(k|k) + \boldsymbol{\xi}_i$, where $\mathcal{N}(G_i B_i(k))$ is the right null space of $G_i B_i(k)$.

Proof: Define $\boldsymbol{\xi}_i = \tilde{\boldsymbol{u}}_i(k|k) - M_i(k) \hat{\boldsymbol{v}}_i(k|k)$. Since $M_i(k)$ is the right inverse of $G_i B_i(k)$ the result follows. ∎

Proposition 4: Let $M_i(k)$ be a right inverse matrix of $G_i B_i(k)$. The optimization problem in (37) is equivalent to the following

$$\begin{aligned}
&\min_{\mathcal{V}_1, \Xi} \sum_{i=1}^S \left(\boldsymbol{c}_{i,u}^T(k) M_i(k) \hat{\boldsymbol{v}}_i(k|k) + \boldsymbol{c}_{i,u}^T(k) \boldsymbol{\xi}_i \right) \\
&\textbf{s.t.} \\
&\underline{\boldsymbol{u}}_i \leq M_i(k) \hat{\boldsymbol{v}}_i(k|k) + \boldsymbol{\xi}_i \leq \overline{\boldsymbol{u}}_i, \qquad i = 1, \dots S \\
&H_i(M_i(k) \hat{\boldsymbol{v}}_i(k|k) + \boldsymbol{\xi}_i) \leq \mathbf{1}, \qquad i = 1, \dots S \\
&F_i B_i(k) M_i(k) \hat{\boldsymbol{v}}_i(k|k) + F_i B_i(k) \boldsymbol{\xi}_i + \\
&\quad \sum_{\substack{j=1 \\ j \neq i}}^S F_{i,z_j} G_j B_j M_i(k) \hat{\boldsymbol{v}}_i(k|k) \leq \overline{\boldsymbol{z}}_i - \mathbf{k}_i(k), \\
&\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, S \\
&\mathcal{V}_1 = \{\hat{\boldsymbol{v}}_1(k|k), \dots, \hat{\boldsymbol{v}}_S(k|k)\} \\
&\Xi = \{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_S\} \\
&\boldsymbol{\xi}_i \in \mathcal{N}(G_i B_i(k)) \qquad\qquad i = 1, \dots, S.
\end{aligned} \qquad (43)$$

Proof: Due to Prop. 3, any feasible point $\hat{\boldsymbol{u}}(k|k)$ for (37) can be written as a feasible point $[\hat{\boldsymbol{v}}(k|k)\}, \boldsymbol{\xi}]$ for (43) with the same cost. Similarly for any feasible point $[\hat{\boldsymbol{v}}(k|k), \boldsymbol{\xi}]$ exists a feasible point $\hat{\boldsymbol{u}}(k|k)$ for (37) which leads to the same cost. ∎

Proposition 4 implies that there always exists a collection of right inverse matrices $M_i(k)$ of $G_i B_i(k)$, $i = 1, \dots, S$ such that the minimum cost of (38) equals the minimum cost of (39). Let $\mathcal{M}_i^\star(k)$ be the set of right inverse matrices of $G_i B_i(k)$ such that when $M_i(k) \in \mathcal{M}_i^\star(k)$ for $i = 1, \dots, S$ the minimum cost of (38) equals the minimum cost of (39). In general, given a choice of matrices $M_i(k)$, $i = 1, \dots, S$, we cannot test whether or not $M_i(k) \in \mathcal{M}_i^\star(k)$ for $i = 1, \dots, S$. However, a partial characterization of the set $\mathcal{M}_i^\star(k)$ is possible through Prop. 1: if the minimum cost of (38) is strictly greater than the sum of the minimum costs obtained for each of the (39) problems, then at least one of the chosen $M_i(k)$ matrices does not belong to the set $\mathcal{M}_i^\star(k)$. A good selection of $M_i(k)$ matrices is to choose the ones for which the minimum cost of (38) equals the sum of the minimum costs of (39).

V. Simulation Results

We consider a data center composed of 6 racks, each having 42 servers and 3 CRAC units. Racks and CRAC units are placed as in Fig. 1. Servers are identical each others and CRAC units are also identical each others. Also, servers have a weak thermal interaction among them and a strong thermal interaction with CRAC units. Jobs are divided among 6 classes, and arrivals are evenly distributed among servers, so that $\lambda_{i_1}^j(k) = \lambda_{i_2}^j(k)$ for all $i_1, i_2 = 1, \dots, 252$ and $j = 1, \dots, 6$. We define this setup as the nominal model.

Figure 2 shows the three relative cost increases for 1000 different data center management problems. Different prob-

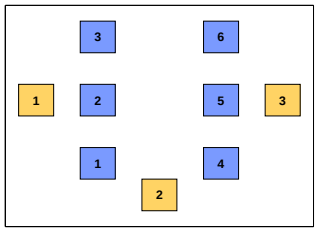

Fig. 1. Data center layout.

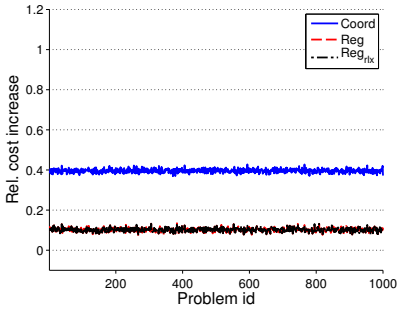

Fig. 2. Relative cost increase, 60% coefficient perturbation.

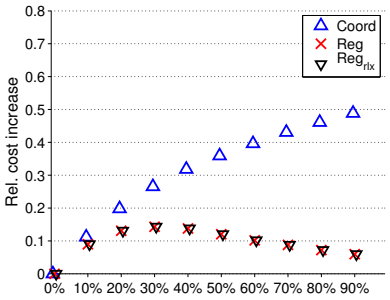

Fig. 3. Relative cost increase for different values of the coefficient of perturbation.

lems were obtained starting from the nominal model and perturbing the parameters randomly within 60%.

The plots in Fig. 2 are: the relative difference between the cost computed by the coordinator controller and the cost of a controller solving (37) (Coord), the relative difference between the sum of the cost obtained by the local regulators and the cost of a controller solving (37) (Reg), and finally, the relative difference between the sum of the cost obtained by the local regulators when solving the optimal problem with weakened constraints discussed in Prop. 2 and the cost of a controller solving (37) (Reg_{rlx}).

Figure 2 shows that the relative cost increase does not change significantly over the different problems. Figure 3 presents the mean cost increase for different values of model perturbation. Each mean point value is computed over 500 different simulations.

For the nominal model, our approach leads to the minimum optimal cost. We observed in our test a difference between the minimum cost computed by the optimal algorithm and our proposed approach on the order of 10^{-5}. The optimality of the proposed algorithm can be explained as follows: the right inverse matrices $M_i(k)$, $i = 1, \ldots, 6$ used in the simulation assumed $\hat{\boldsymbol{u}}_i(k|k) = \frac{\hat{\boldsymbol{v}}(k|k)}{n_i}$. In the simulated data center cases, this implies an even distribution of the computational resources for different job classes, i.e. $\rho_i^{j_1}(k) = \rho_i^{j_2}(k)$. When the data center presents the symmetries described in the nominal model, such a partition of the server resources is the optimal one. In this case then, the chosen inverse matrix set is able to minimize the cost function of the optimal problem in (38) over all possible choices of the inverse matrix set range.

As the value of the coefficient of perturbation increases, the relative difference between the minimum cost found by the coordinator and the one computed by a controller solving (37) increases. When the sum of the costs is obtained by the local regulators instead, their optimal solutions induce a cost function increase of about 10%.

VI. Discussion

This paper presents a control-oriented model of large-scale data centers, including the coupling in the dynamics between the computational resources (servers) and cooling resources (CRAC units). To deal with the size and information distribution within a data center, a hierarchical strategy is proposed in which a higher-level controller computes set-points for aggregated power states for servers in racks. The lower-level controllers solve local optimization problems leading to an improvement in the solution obtained using aggregated variables at the higher level. Our simulations suggest that the approach may be effective, but several research directions should be pursued to fully evaluate the approach and to extend it to more general situations. When the arrivals are quite predictable, it will be useful to be able to optimize the economic parameters over a long temporal horizon.

References

[1] C. Bash, C. Patel, A. Shah, and R. Sharma. The sustainable information technology ecosystem. *ITHERM*, May 2008.

[2] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling "cool": temperature-aware workload placement in data centers. In *ATEC*, Berkeley, CA, USA, 2005. USENIX Association.

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

[4] L. Parolini, N. Tolia, B. Sinopoli, and B. H. Krogh. A cyber-physical systems approach to energy management in data centers. In *ICCPS*, Apr. 2010.

[5] C. D. Patel and A. J. Shah. Cost model for planning, development and operational of a data center. Technical report, Internet Systems and Storage Laboratory, HP Laboratories Palo Alto, Jun. 2005.

[6] R. Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *Journal of Process Control*, 19:723 – 731, 2009.

[7] 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.

[8] U.S. Environmental Protection Agencys. Report to congress on server and data center energy efficiency. Technical report, ENERGY STAR Program, Aug. 2007.