Adaptive Robust Optimization for Coordinated Capacity and Load Control in Data Centers

Xiaoqi Yin and Bruno Sinopoli

Abstract—This paper addresses the problem of improving energy efficiency and quality-of-service (QoS) of data centers, by coordinating the “feed-forward” capacity provisioning controller and the “feed-back” load balancing controller. A data center is modeled as a collection of modular server blocks which cooperatively process multi-class, inter-dependent workload. We propose a coordinated two-stage control strategy of data centers based on the adaptive robust optimization framework. In stage 1, the optimal capacity of each server block is found based on predicted arrival rates of future workload, taking into account the potential QoS cost in stage 2; Then in stage 2, the load balancer distributes incoming workload to server blocks to achieve optimal QoS, after observing the actual workload. We show through simulations that the proposed approach achieves lower total costs as well as less QoS variations compared to a start-of-art baseline approach with reasonable level of conservativeness.

I. INTRODUCTION

Data centers consume a significant amount of energy today. According to U.S. Environmental Protection Agency, U.S. data centers consumed 1.5 percent of total U.S. electricity generation in 2007 [4]. However, the way data centers use energy is far from efficient. To this end, much research effort has been made on improving data center energy efficiency without compromising the quality-of-service (QoS).

Capacity provisioning and load balancing are among the most widely adopted approaches in data center control [11]. The capacity provisioning approach adjusts the number of active servers in the data center to match the total computing power with the total workload to be processed, by switching on/off servers [7]. However, switching servers on/off is time consuming and therefore needs to be well planned ahead of time in a “feed-forward” way, based on the predicted amount of workload, electricity prices, as well as cooling efficiency. In cases where the prediction error is high, the performance of capacity provisioning can be degraded.

On the other hand, the load balancing approach distributes the incoming workload to servers in an intelligent way to minimize the energy and QoS cost [9]–[11]. In contrary to capacity provisioning, a load balancer can react to the current workload instantly, and thus can serve as an “feedback” controller without the need of knowing future workload.

In this paper, we propose a novel coordinated control strategy of data centers based on the adaptive robust optimization framework. The proposed controller minimizes the energy cost and the worst-case QoS cost under stochastic workload by coordinating the “feed-forward” capacity provisioning controller and the “feed-back” load balancing controller. The key observation behind the proposed approach is that while the performance of capacity provisioning suffers from prediction error of future workload, the load balancer can act after observing the actual workload to improve control performance. By optimally coordinating and allocating control efforts, better performance can be achieved even under prediction error of workload.

The contribution of this paper is three-fold: First, we develop a novel and extendable model of data centers capturing the inter-dependent nature of multi-class workload and the modular nature of data center design. In addition to prior modeling of multi-class workload coupled only in terms of resource requirements [10], our model also considers inter-dependence of their arrival rates, which is aligned with the observation that modern IT services can consist of multiple inter-dependent functional components [1]. The model can also be naturally extended to consider changing electricity prices and locationally varying cooling costs.

Second, we develop a novel coordinated control strategy for addressing energy vs. QoS tradeoff in data centers based on adaptive robust optimization. Coordinated control of capacity provisioning and load balancing has been studied previously by [6], [10]. However, [6] only considers static power/load allocation strategy; [10] considers the dynamic allocation but assumes the future workload is known. Our proposed approach assumes that future workload can be predicted with uncertainty, and incorporate this uncertainty into controller design. We adopt adaptive robust optimization framework, which is a recently developed approach for two-stage optimization problems under uncertainty [2], [3]. To our best knowledge, this paper is the first to apply adaptive robust optimization methods in data center control problem.

Third, we validate the proposed approach using realistic workload traces. Simulation results show that the proposed approach can achieve less expected total energy and QoS cost, as well as reduced cost variations, compared to baseline approach. We also observe an interesting tradeoff between the expected cost and the variation of costs, which can be controlled by varying the level of conservativeness.

The rest of the paper is structured as follows: In Section II we develop the model for workload and data center; In Section III, we introduce and formulate the proposed coordinated control strategy, then provide algorithms for solving the adaptive optimization problem; Simulation results is provided in Section IV; In Section V, we suggest future directions and conclude the paper.

1 Xiaoqi Yin and Bruno Sinopoli are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA xiaojiy@andrew.cmu.edu, brunos@ece.cmu.edu
II. MODELING

In this section, we develop mathematical model of the workload and the data center. Figure 1 provides an architectural view of the model of the workload and data center, as well as controllers. We elaborate each of the components in more details as follows.

A. The Workload Model

The time horizon of interest is divided into a set of consecutive time slots with duration $\Delta t \in \mathbb{R}_+$, denoted as $T = \{1, 2, \cdots, T\}$. In practice, the duration $\Delta t$ can be as long as 5 minutes, and $T$ can be 12, which creates a 1-hour time horizon. We use $t = 0$ to denote the initial stage before the time horizon of interest starts.

We assume that there are $J$ classes of jobs. Let $\mathcal{J} = \{1, \cdots, J\}$ be the set of job classes. Each class of jobs $j \in \mathcal{J}$ arrive at the data center as an independent poisson process with arrival rate $\lambda_j(t) \in \mathbb{R}_+$ during time slot $t$. Let $\lambda(t) = (\lambda_1(t), \cdots, \lambda_J(t))^T$. Note that a server job arrival rate at server block $i$ during time slot $t$. Let $\lambda_S(t) = (\lambda_S^n(t), \cdots, \lambda_S^S(t))^T$. Note that a server job arrival rate at server block $i$ during time slot $t$. Let $\lambda_S(t) = (\lambda_S^n(t), \cdots, \lambda_S^S(t))^T$. Note that a server

B. The Data Center Model

We model the data center as a set of server blocks, $S = \{1, \cdots, S\}$. A server block $i$ is a set of $M_i$ homogeneous servers, physically located in a region and controlled by a virtual machine. By “homogeneous” we mean that all servers in block $i$ has the same job processing rate as well as power consumption characteristic. We consider server blocks as basic components of a data center to capture the modular nature of data centers.

Let $x_i(t) \in \mathbb{N}$ be the number of active servers in server block $i$, $0 \leq x_i(t) \leq M_i$. Since $M_i$ is usually as large as $10^3$, in the rest of the paper, we let $x_i(t) \in \mathbb{R}_+$ for mathematical simplicity. To keep notation compact, we use $\Omega_x$ to denote the set of all feasible number of active servers. Note that $x_i(0)$ is the initial number of active servers, which is given at $t = 0$. Let $x(t) = (x_1(t), \cdots, x_S(t))^T$, $x \in \{x(t), t = 1, \cdots, T\}$. The number of active servers in a server block can be adjusted by switching on/off servers, which is called capacity provisioning. However, switching on/off servers takes time as well as energy. We assume that it takes one time slot to switch on/off a server. The energy cost to switch on/off one server in server block $i$ is $\beta_i x_i(t)$. Thus, the total energy cost incurred by switching on/off servers in server block $i$ at the beginning of time slot $t$ is $\beta_i [x_i(t+1) - x_i(t)]$.

Let $e_i(t, x_i(t))$ be the energy cost of active servers in block $i$ during time slot $t$, given number of active servers $x_i(t)$. Note that $e_i(t, x_i(t))$ may include the cost of electrical energy used by server block $i$ for computing, as well as cost of energy used for cooling the server block $i$ to desired temperature [8]. One example of $e_i(\cdot, \cdot)$ is that $e_i(t, x_i(t)) = E_i(t) x_i(t) + C_i(t) x_i(t)$, where $E_i(t) \in \mathbb{R}_+$ is a predefined parameter capturing the electrical energy consumption of one server in block $i$ in one time slot, as well as time-varying electricity prices, $C_i(t) \in \mathbb{R}_+$ is a cooling coefficient, which varies over time depending on external factors, such as weather conditions, etc. Thus, the total energy cost of server block $i$ over the entire time horizon is defined as

$$\sum_{i=1}^{T} e_i(t, x_i(t)) \geq \beta_i [x_i(t+1) - x_i(t)]$$

Job Processing Model. As shown in Figure 1, the incoming jobs first arrive at the load balancer, and is then directed by the load balancer to each server blocks. Let $\lambda^S(t) \in \mathbb{R}_+$ be the aggregated job arrival rate at server block $i$ during time slot $t$. Let $\lambda^S(t) = (\lambda^S_1(t), \cdots, \lambda^S_S(t))^T$. Note that a server
block may receive more than one classes of jobs, depending on the control strategy of the load balancer. Let \( d_i^S(x_i(t), \lambda_i^S(t)) \) be the mean response time of all jobs processed at server block \( i \) during time slot \( t \), given number of active servers \( x_i(t) \) and arrival rate \( \lambda_i^S(t) \). Let \( d^S(t) = (d_1^S(t), \ldots, d_J^S(t))^T \). We assume that \( d_i^S(x_i(t)) \) is increasing in \( \lambda_i(t) \), decreasing in \( x_i(t) \), and convex in both variables. In this paper, we assume that all jobs are fully parallelizable within a server block, and the processing rate variables. In this paper, we assume that all jobs are fully parallelizable within a server block, and the processing rate variables.

**Load Balancing Model.** The goal of the load balancer is to optimally direct incoming jobs to server blocks so as to maximize QoS. Let \( q_{ij}(t) \in [0, 1] \) be the fraction of class \( j \) jobs assigned to server block \( i \) by the load balancer during time slot \( t \). Then the matrix \( Q(t) = \{q_{ij}(t)\}_{i \in S, j \in J} \) is called load balancing matrix. Note that \( \sum_{i \in S} q_{ij}(t) = 1, \forall j \in J, t = 1, \ldots, T \).

Also, we assume that different classes of jobs have different resource requirements, and can only be processed by a subset of all server blocks. Let \( S_j \subseteq S \) be the set of blocks which can process class \( j \) jobs. Then, we have that \( q_{ij}(t) = 0, \forall i \notin S_j, \forall j \in J, t = 1, \ldots, T \). Let \( \Omega_q \) be the set of all feasible load balancing matrices to keep notation compact.

The load balancer takes incoming jobs of all classes, and distributes the jobs to server blocks according to the following rule:

\[
\lambda^S(t) = Q(t)\lambda(t)
\]  

(3)

We also note that the following relationship exists between mean response time of a particular job class \( j \), \( d_j(t) \), and the mean response time of jobs processed by block \( i \), \( d_i^S(t) \):

\[
d(t) = Q^T(t)d^S(t)
\]  

(4)

Note that given (1),(3),(4), the QoS cost of job class \( j \) during time slot \( t \), \( c_j \), can be written as a function of \( x(t) \), \( \lambda(t) \) and \( Q(t) \), namely,

\[
c_j = c_j(x(t), \lambda(t), Q(t))
\]  

(5)

To sum up, we developed in this section a model for data centers, which can be controlled by 1) adjusting number of active servers in all server blocks, \( x(t) \); 2) adjusting the load balancing matrix \( Q(t) \). Given \( x(t) \) and \( Q(t) \), the total energy cost of the entire time horizon is \( J_{\text{energy}}(x) = \sum_{t=1}^T e_i(t, x_i(t)) + \beta_i |x_i(t + 1) - x_i(t)| \), while the total QoS cost is \( \sum_{t=1}^T \sum_{j \in J} c_j(x(t), \lambda(t), Q(t)) \). We show the architecture of the model in Figure 1.

**III. THE PROPOSED DATA CENTER COORDINATED CONTROL STRATEGY**

An overview of the proposed controller is shown in Figure 2 and Table I. The proposed coordinated control strategy of the data center consists of two stages:

- **Stage 1:** Capacity provisioning. At time \( t = 0 \), the capacity controller obtains the capacity of the data center over the entire horizon \( x^*(\Omega) \), given the region of possible future arrival rates, \( \Omega_{\lambda} \). Note that the decision is made by taking into consideration the potential worst-case QoS cost that can be achieved by load balancer which acts after observing the actual job arrival rates \( \lambda \).
- **Stage 2:** Load balancing. At time \( t = 1, 2, \ldots, T \), given capacity of the data center \( x^*(t) \) decided in stage 1, the load balancer observes the actual job arrival rates \( \lambda(t) \), and then obtains optimal load balancing matrix \( Q^*(x^*(t), \lambda(t)) \) in response to the observed arrival rates so as to achieve minimum QoS cost.

In the rest of this section, we first investigate the optimal stage 2 load balancing strategy \( Q^*(x^*(t), \lambda(t)) \), given capacity \( x^*(t) \) obtained in stage 1 and observed arrival rates \( \lambda(t) \). Then we introduce the concept of worst-case stage 2...
QoS cost, and formulate the optimization problem to obtain optimal stage 1 strategy $x^*(\Omega_\lambda)$.

### A. Optimal Load Balancing Strategy in Stage 2

The load balancing controller works in a feedback way, namely, at time $t$, it obtains optimal load balancing matrix $Q^*(t)$, after observing the arrival rates $\lambda(t)$. $Q^*(t)$ is computed by solving the following optimization problem:

$$Q^*(x(t), \lambda(t)) = \arg \min_{Q(t)} \sum_{j \in J} c_j(x(t), \lambda(t), Q(t)) \tag{6a}$$

subject to:

$$Q(t) \lambda(t) \geq x(t) \tag{6b}$$

$$Q(t) \in \Omega_Q, \ \forall t \in \mathcal{T} \tag{6c}$$

The optimal value of the objective function, denoted as $J_{QoS}(x(t), \lambda(t))$, is the minimum QoS cost that can be achieved in stage 2 during time slot $t$, given $x(t)$ and $\lambda(t)$. Problem (6) is a convex optimization problem, given $x(t)$ and $\lambda(t)$ as inputs. One can show that $J_{QoS}(x(t), \lambda(t))$, is also convex. While in general optimization problem (6) must be numerically solved to obtain $Q^*(x(t), \lambda(t))$, in the following particular case, part of $Q^*(x(t), \lambda(t))$ can be explicitly expressed, as shown in the following theorem.

**Theorem 1:** Suppose for job class $j \in \mathcal{J}$, $\mathcal{S}_j \cap \mathcal{S}_k = \emptyset$, $\forall k \in \mathcal{J} \setminus \{j\}$. Then given $x_i(t), \forall i \in \mathcal{S}_j$, the optimal load balancing strategy for job class $j$ during time slot $t$ is given by

$$q^*_j(t) = \frac{x_i(t)}{\sum_{i \in S_j} x_i(t)}$$

The intuition behind this theorem is that if a particular job class can only be processed by a subset of server blocks, and this set of server blocks cannot process other classes of jobs, then the optimal load balancing strategy for this class of jobs is to assign workload to blocks in proportion to their numbers of active servers.

### B. Optimal Capacity Provisioning Startegy in Stage 1

In this section, we proposed an approach to obtain capacity provisioning strategy $x^*(\Omega_\lambda)$ based on adaptive robust optimization. We first define worst-case QoS cost in stage 2. Then an adaptive robust optimization problem is formulated to find $x^*(\Omega_\lambda)$ so as to minimize energy cost as well as worst-case QoS cost.

**Worst-Case QoS Cost.** As shown in previous section, given $x(t)$ and $\lambda(t)$, the optimal QoS cost during time slot $t$ is $J_{QoS}(x(t), \lambda(t))$. However, capacity provisioning decisions, $x$, are made at time $t = 0$, at which point the true arrival rates $\lambda(t), t = 1, \cdots, T$ is not known. However, the only information of the future workload known at time $t = 0$ is the region of possible future arrival rates $\Omega_\lambda$. A method is then needed to estimate at time $t = 0$ the potential QoS cost given $\Omega_\lambda$. In this paper, we define the potential QoS cost as the worst-case total QoS cost that can be achieved by possible future job arrivals in $\Omega_\lambda$:

$$\pi_{wc}^{QoS}(x; \Omega_\lambda) = \max_{\lambda \in \Omega_\lambda} \sum_{t=1}^{T} J_{QoS}(x(t), \lambda(t)) \tag{7}$$

Note that the worst-case cost is not the only possible formulation of potential QoS cost when the future arrival rates is stochastic. An alternative approach is to consider expected QoS cost. However, the reason why we use the concept of worst-case QoS cost is two-fold: First, while energy cost of data center has received increasing attention, the QoS is of primary concern for the data center operators. Thus, the worst-case QoS cost formulation is consistent with the conservativeness of the data center operations in practice. Second, the worst-case cost formulation makes the stage 1 optimization problem more mathematically tractable, while computationally expensive Monte Carlo simulation is usually needed to evaluate the expected cost.

Note that if $\Omega_1 \subset \Omega_2$, then $\pi_{wc}^{QoS}(x; \Omega_1) \leq \pi_{wc}^{QoS}(x; \Omega_2)$. Thus, one can adjust the size of the uncertainty set $\Omega_\lambda$ to tune the conservativeness of the controller.

**Proposed Stage 1 Stategy.** At time $t = 0$, the following optimization problem is solved to obtain optimal capacity provisioning strategy in stage 1, taking into consideration the worst-case QoS cost.

$$x_{\text{robust}}^*(\Omega_\lambda) = \arg \min_x J_{\text{energy}}(x) + \max_{\lambda \in \Omega_\lambda} \sum_{t=1}^{T} J_{QoS}(x(t), \lambda(t)) \tag{8a}$$

subject to:

$$x(t) \in \Omega_x, \ \forall t \in \mathcal{T} \tag{8b}$$

At this point, we introduce two alternative stage 1 strategy to compare with the proposed strategy based on adaptive robust optimization, namely, offline optimal stage 1 strategy and baseline stage 1 strategy. We show that both of them are special cases of the proposed strategy (8). In Section IV, the performance of all methods are compared through numerical simulations.

**Offline Optimal Stage 1 Stategy.** The offline optimal stage 1 decisions are defined as the optimal capacity decisions $x_{off}(\lambda)$ with perfect knowledge of the actual future arrival rates over the entire time horizon $\lambda$. In particular, it is a special case of the proposed robust approach with
uncertainty set $\Omega_\lambda$ degenerated to a singleton $\{\lambda\}$:

$$x^\ast_{\text{offline}}(\lambda) = x^\ast_{\text{robust}}(\{\lambda\}) \quad (9)$$

As offline optimal strategy assumes perfect knowledge of future, the resulting cost provides a lower bound of achievable total costs by any algorithms.

**Baseline Stage 1 Strategy.** In practice, the actual arrival rates $\lambda$ is not known at time $t = 0$, but can be predicted as $\hat{\lambda}$. One commonly adopted heuristic is to regard the predicted $\lambda$ as accurate and obtain capacity decisions by:

$$x^\ast_{\text{baseline}}(\hat{\lambda}) = x^\ast_{\text{robust}}(\{\hat{\lambda}\}) \quad (10)$$

We refer the solution $x^\ast_{\text{baseline}}(\hat{\lambda})$ as the baseline stage 1 strategy. In cases where prediction error is present, the solution is clearly sub-optimal. Note that the baseline stage 1 strategy is also a special case of the proposed strategy when $\Omega_\lambda = \{\lambda\}$.

**C. Algorithms for Solving Stage 1 Problem.**

While the proposed stage 1 problem (8) is a convex optimization problem, the main difficulty for solving the problem is that the worst-case QoS cost term, $J_{QoS}^{nc}(x; \Omega_\lambda)$, is hard to evaluate. This is because given $x$, computing the value of $J_{QoS}^{nc}(x; \Omega_\lambda)$ involves solving an optimization problem (7). To tackle this difficulty, we use Benders Decomposition type method proposed in [3] to solve the problem. The basic idea of the algorithm is to use piecewise linear function to iteratively approximate $J_{QoS}^{nc}(x; \Omega_\lambda)$. We provide the outline of the algorithm as follows, and refer the readers to [3] for more details.

**Algorithm for Solving Stage 1 Problem.**

1) Initialize: Set lower bound of objective function $L = -\infty$, upper bound $U = \infty$. Let $x_0$ be an initial feasible solution to problem (8). Choose error tolerance level $\epsilon \in \mathbb{R}_+$. Set iteration counter $k = 1$.

2) Solve the master problem:

$$\min_{x, \gamma} J_{\text{energy}}(x) + \gamma \quad (11)$$

s.t. $x(t) \in \Omega_x$, $\forall t \in T$

$$\gamma \geq J^n + (\mu^n)^T(x - x^n), \quad \forall n < k$$

The optimal solution of this problem is denoted as $(x_k, \gamma_k)$. Set $L$ to be the optimal objective value.

3) Solve the subproblem:

$$J_{QoS}^{nc}(x^k; \Omega_\lambda) = \max_{\lambda \in \Omega_\lambda} \sum_{t=1}^{T} J_{QoS}(x^k(t), \lambda(t)) \quad (12)$$

and obtain a) the optimal objective value $J^k$; b) the sensitivity of $J^k$ with respect to $x$ at $x = x^k$, denoted as $\mu^k$. Then, add constraint

$$\gamma \geq J^k + (\mu^k)^T(x - x^k)$$

to the master problem. Set $U = J_{\text{energy}}(x^k) + J^k$.

4) Check if $U - L \leq \epsilon$, stop, return $x_k$ as solution; otherwise, go to Step 2, and set $k = k + 1$.

As the problem (8) is convex, the algorithm converges to the optimal solution of problem (8) in finite steps [5].

**IV. SIMULATION RESULTS**

In this section, we empirically validate the proposed approach through simulations based on realistic workload traces. While one can conduct full-scale evaluation by specifying sophisticated detailed models of workload and data centers, we make simplifying assumptions to provide an intuitive proof-of-concept of the proposed approach.

**A. Experiment Setup**

**Workload.** We consider a time horizon of 1-hour divided into 12 of 5-minute slots, and a total of 3 job classes. The workload traces are generated based on the real-world arrival rates of Google products [1] in a peak hour. Figure 3 shows the expected arrival rate traces of workload.

In order to empirically evaluate the performance of the proposed approach, we randomly generate 1000 traces of actual arrival rates based on the predefined uncertainty set. In particular, we assume that the actual arrival rates, $\lambda(t)$, are uniformly distributed in the uncertainty set $\Omega_\lambda(t)$ defined in (2), namely, $\lambda(t) \sim \text{uniform}(\Omega_\lambda(t))$, where $\alpha = 1$. We let $\Delta \lambda_j(t) = 0.1 \lambda_j(t)$.

The QoS cost function is defined as $c_j(\lambda_j(t), d_j(t)) = C_j \lambda_j(t) d_j(t), \ \forall j \in \mathcal{J}$. We let $C_1 = 1, C_2 = 0.8, C_3 = 0.6$. Namely, class 1 jobs have the highest sensitivity to delays, while class 3 is the most delay-tolerant.

**Data Center Parameters.** We assume that there are 3 server blocks in the data center, each of which consists of 5000 servers with the same settings. Each block can only process one class of workload: $S_1 = \{1\}, S_2 = \{2\}, S_3 = \{3\}$. We let active energy cost $e_i(t, x_i(t)) = E_i x_i(t), E_i = 1, \forall i \in S$. For switching cost, we let $\beta_i = 1, \forall i \in S$.

**Performance Metrics.** Given a particular capacity decision $x$ and actual job arrival rate $\lambda$, the corresponding total cost is $J(x, \lambda) = J_{\text{energy}}(x) + \sum_{t=1}^{T} J_{QoS}(x(t), \lambda(t))$. We define normalized cost by

$$nJ(x, \lambda) = \frac{J(x, \lambda)}{J(x^\ast_{\text{offline}}(\lambda), \lambda)} \quad (13)$$

Since both job arrival rate $\lambda$ and the resulting normalized cost $nJ(x, \lambda)$ are stochastic, we use the expected normalized cost $E_\lambda(nJ(x, \lambda))$ as well as the standard deviation of the normalized cost $\text{std}(nJ(x, \lambda))$ to evaluate the control approaches.

**B. Performance Analysis**

We first compare the expected total cost achieved by the proposed and baseline approaches in Figure 4a. The $\alpha$ is the parameter controlling the size of the uncertainty set $\Omega_\lambda$ and thus the conservativeness of the proposed approach. A larger $\alpha$ leads to more risk-averse control decisions.

When $\alpha = 0$, $\Omega_\lambda(t) = \{\hat{\lambda}(t)\}$, the proposed approach is equivalent to the baseline approach. As $\alpha$ starts to increase, more and more conservativeness is inserted into the decision making of the proposed approach, which leads to reduced expected normalized cost. However, if $\alpha$ grows too large, the expected normalized cost achieved by the proposed
approach starts to increase, and can become even higher than the baseline approach, due to the over-conservativeness it encodes into the solution: More servers are turned on unnecessarily to get ready for the worst-case arrival of jobs, which results in higher cost. There is an optimal $\alpha^*$ and corresponding level of conservativeness, which is achieved at $\alpha^* = 0.63$ in this test scenario. At this point, the proposed approach achieves the minimum expected normalized cost, which is around 15% less than that of the baseline approach.

While there is no theoretical rule to obtain optimal $\alpha$ for a particular setting of the problem, this parameter $\alpha$ can be chosen empirically and tuned online to achieve better performance of the proposed approach for practical use.

In addition to expected total cost, we also compare the variability of normalized costs achieved by proposed approach and the baseline approach, as shown in Figure 4b. We observe that the standard deviation of normalized cost achieved by proposed approach is lower than that achieved by the baseline approach. Also, the variability of the proposed approach reduces as $\alpha$ increases. Therefore, by using the proposed approach, lower variability of QoS can be achieved, which is aligned with the risk-averse nature of the worst-case formulation of QoS costs.

This leads to an interesting tradeoff between the expected cost and the variability of costs, controlled by parameter $\alpha$: When $\alpha \geq \alpha^*$, the smaller expected cost is achieved at the cost of high QoS variability. In cases where high QoS variability is undesirable, the data center operator must accept higher expected cost when tuning $\alpha$ to reduce QoS variability. Being aware of this cost vs. variability tradeoff, the parameter $\alpha$ can be chosen by the data center operators based on their preferences in practice.

V. CONCLUSION

This paper considers the problem of controlling data centers for reducing energy and QoS cost. We develop a mathematical model of the data center and workload, capturing the modular nature of data centers and the inter-dependent nature of the multi-class workload. We then proposed a coordinated two-stage control strategy based on adaptive robust optimization. Simulation results show that by carefully selecting the size of uncertainty set, the proposed approach can achieve better performance comparing to the baseline approach, in terms of expected total cost as well as QoS variability.

The future work is two-fold: First, the framework proposed in this paper can be extended to study how data centers optimally participate in demand-response program in smart grid; Second, it is natural to design a distributed approach for capacity provisioning and load balancing based on the proposed framework.

REFERENCES