Robust Heterogeneous Data Center Design: A Principled Approach

Siddharth Garg
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada

Shreyas Sundaram
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada

Hiren D. Patel
Electrical and Computer Engineering
University of Waterloo
Waterloo, ON, Canada

1. INTRODUCTION

Data centers represent the fastest growing component of information and communication technologies (ICT) energy footprint. With the advent of cloud computing, data centers will increasingly be used to process a wide array of jobs with differing characteristics such as degree of parallelism, memory access patterns *etc.*. From an energy efficiency perspective, the most energy efficient server architecture differs for jobs with different characteristics [4], motivating the need to consider *heterogeneous* data center designs consisting of many server types [3, 5].

Even though types of jobs that a data center is expected to serve might be known at design time, the workload statistics are often unknown until the data center is deployed. Therefore, data centers should be designed keeping in mind the uncertainty in workload statistics — in this paper, we outline a principled approach to *designing* energy-efficient, heterogeneous data centers that are robust against data center workload variations, using Wald's minimax criterion as a starting point. In the proposed formulation, we assume that the only thing that is known at design time is an upper bound on the total rate (over all job types) at which jobs arrive at the data center, and design the data center to have the minimum *worst-case* energy consumption over all job type mixes. We then highlight a number of potential avenues for further investigation.

2. PROBLEM FORMULATION

We assume that there N types of commercially available *server architectures* that can be used to design the data center, and that the data center is expected to serve M different *job types* that are pre-characterized. The power consumed while executing an instance of a job of type j $(1 \leq j \leq M)$ on a server of type i $(1 \leq i \leq N)$ is given by e_{ij}, and the corresponding service rate is given by μ_{ij}. Furthermore, the idle mode power consumption of server i is given by e_i^{idle}.

We assume that per-type jobs arrivals are modeled as independent point Poisson processes, and that the arrival rate of jobs of type j is given by λ_j. The vector $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_M]^T$ is referred to as the *arrival rate vector*. To reflect the uncertainty in workload statistics at design time, we assume that the individual arrival rates for each job type are *not* known a priori. Instead, we only know an upper bound on the total arrival rate of all jobs into the data center, i.e., $\sum_{j\in[1,M]} \lambda_j \leq \lambda_{max}$.

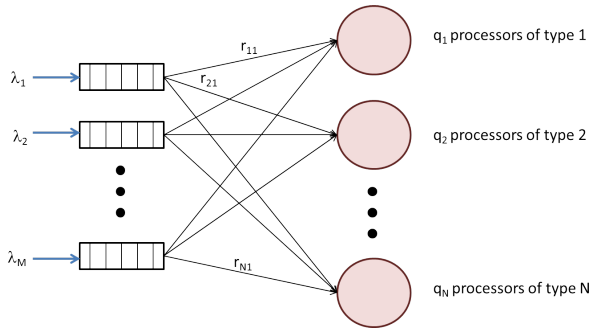

Figure 1: Problem set-up

We denote by q_i the number of servers of type i that are included in the data center. The vector $Q = [q_1, q_2, \ldots, q_N]^T$ is referred to as the *data center design vector*. Each server has an associated cost, for example, its market price, c_i, and the data center must be designed within a total budget given by c_{budget}.

Finally, we assume that there exists a data center resource manager (RM) that maps incoming jobs to servers at runtime. We assume that the RM can distinguish between job types and slots them into one of M virtual queues, one for each job type. The RM schedules the execution of jobs in each virtual queue to a subset of available servers in the data center such that the queues remain stable, i.e., the service rate of each queue matches (or exceeds) its arrival rate. The matrix $R \in \mathbb{R}^{N\times M}$ is called the *data center scheduling matrix*, and element r_{ij} indicates the number of servers of type i that are used to process jobs of type j. The problem set-up is illustrated in Figure 1. It is interesting to note that the model described here is an generalization of the data center modeled discussed by [2], where the authors assume a homogeneous data center with single job and server types and a known job arrival rate.

The goal of the robust heterogeneous data center design problem is to find the optimal data center design vector Q^*, that *minimizes the worst-case power consumption over all admissible arrival rate vectors, assuming that for any data*

center design vector and arrival rate vector, the RM makes optimal scheduling decisions.

2.1 Minimax Optimization

We begin with some definitions for ease of exposition.

Definition 1. Given an arrival rate vector Λ and data center design vector Q, $R_S(Q,\Lambda)$ is defined to be the set of all RM scheduling decisions that ensure that the queues remain stable. More specifically, $R_S(Q,\Lambda)$ is the feasible region of the following set of linear inequalities:

$$\sum_{i\in[1,N]} \mu_{ij} r_{ij} \geq \lambda_j \quad \forall j \in [1,M]$$

$$\sum_{j\in[1,M]} r_{ij} \leq q_i \quad \forall i \in [1,N]$$

$$r_{ij} \geq 0 \quad \forall i \in [1,N], \forall j \in [1,M]$$

Definition 2. $E(Q,\Lambda,R)$ is defined to be the power consumption of a data center for a given data center design vector Q, arrival rate vector Λ, and data center scheduling matrix R.

$$E(Q,\Lambda,R) = \begin{cases} \sum_i \sum_j r_{ij}(e_{ij} - e_i^{idle}) + q_i e_i^{idle} & R \in R_S(Q,\Lambda) \\ \infty & \text{otherwise} \end{cases}$$

We begin with by formulating a deterministic problem in which both arrival rate vector Λ and data center design Q are known, and the goal is to determine the optimal scheduling matrix, $R^*(Q,\Lambda)$, that minimizes the data center power consumption.

LP-Deterministic.

$$E^*(Q,\Lambda) = \min_R E(Q,\Lambda,R) \tag{1}$$

subject to:

$$R \in R_S(Q,\Lambda)$$

The robust heterogeneous data center design problem can now be written as:

Minimax.

$$\min_Q \max_\Lambda E^*(Q,\Lambda) \tag{2}$$

subject to:

$$\sum_{i\in[1,M]} \lambda_i \leq \lambda_{max}$$

$$\sum_{i\in[1,N]} c_i q_i \leq c_{budget}$$

To solve the minimax problem, we first define some additional terminology.

Definition 3. Λ^i $(i \in [1,M])$ represents an arrival rate vector in which jobs of type i arrive at rate λ_{max} and all other jobs arrive at rate 0. Formally, $\lambda_j^i = 0$ if $j \neq i$ and $\lambda_j^i = \lambda_{max}$ if $j = i$.

LEMMA 1. *Let Λ' be any admissible arrival rate vector, i.e., $\sum_{i\in[1,M]} \lambda_i' \leq \lambda_{max}$, then:*

$$E^*(Q,\Lambda') \leq \max_{i\in[1,M]} (E^*(Q,\Lambda^i))$$

.

PROOF. We begin by noting that for any Λ', there exists a vector $\Delta = [\delta_1, \delta_2, \ldots, \delta_M]^T$ such that $\Lambda' = \sum_{i\in[1,M]} \delta_i \Lambda^i$, where $\delta_i \geq 0$ $(\forall i \in [1,M])$ and $\sum_{i\in[1,M]} \delta_i \leq 1$.

Now define $R' = \sum_{i\in[1,M]} \delta_i R^*(Q,\Lambda^i)$. It can be shown that R' is a feasible solution for data center design vector Q and arrival rate Λ', i.e., $R' \in R_S(Q,\Lambda')$.

In addition, we can show that:

$$E(Q,\Lambda',R') = \sum_{i\in[1,M]} \delta_i E^*(Q,\Lambda^i)$$

.

By definition, $E^*(Q,\Lambda') \leq E(Q,\Lambda',R')$ and given that $\sum_{i\in[1,M]} \delta_i E^*(Q,\Lambda^i) \leq \max_{i\in[1,M]}(E^*(Q,\Lambda^i))$, the desired result is obtained. □

Lemma 1 allows us to simplify the Minimax problem as follows:

Minimax.

$$\min_{Q,\gamma} \gamma \tag{3}$$

subject to:

$$E^*(Q,\Lambda^i) \leq \gamma$$

$$\sum_{i\in[1,N]} c_i q_i \leq c_{budget}$$

We will now determine an analytical expression for $E^*(Q,\Lambda^t)$ $(t \in [1,M])$. Recall that $E^*(Q,\Lambda^t)$ represents the minimum data center power consumption given a data center design Q and assuming that jobs of only type t arrive at the data center, and the rate at which they arrive is λ_{max}. We will assume, for notational simplicity and without any loss of generality, that for job type t, the following relationship holds:

$$\frac{e_{it} - e_i^{idle}}{\mu_{it}} \leq \frac{e_{jt} - e_i^{idle}}{\mu_{jt}} \quad \forall i,j \in [1,N]; j \geq i \tag{4}$$

Note that $\frac{e_{it}-e_i^{idle}}{\mu_{it}}$ can be viewed, in a sense, as a measure of the *energy efficiency* of servers of type i in processing jobs of type t. This is becomes more clear if we set $e_i^{idle} = 0$. Equation 4 simply says that for jobs of type t, the servers are indexed in decreasing order of energy efficiency[1]. Under this assumption, we can write an expression for $E^*(Q,\Lambda^t)$.

Definition 4.

$$e_{it} - e_i^{idle} = e_{it}^{diff} \quad \forall i \in [1,N]$$

LEMMA 2.

$$E^*(Q,\Lambda^t) = \max_{j\in[1,M]} \left(\sum_{i=1}^{j-1} q_i e_{it}^{diff} + \frac{\lambda_{max} - \sum_{i=1}^{j-1} q_i \mu_{it}}{\mu_{jt}} e_{jt}^{diff} \right) \tag{5}$$

[1]Similar results can be derived for every other task type by appropriately re-ordering the indices in decreasing order of energy efficieny.

.

PROOF. We can obtain $E^*(Q, \Lambda^t)$ by solving the following linear program:

$$E^*(Q, \Lambda^t) = \min_R \sum_{i \in [1,N]} (r_{it}(e_{it} - e_i^{idle}) + q_i e_i^{idle}) \quad (6)$$

subject to:

$$\sum_{i \in [1,N]} \mu_{it} r_{it} \geq \lambda_{max}$$

$$r_{it} \leq q_i \quad \forall i \in [1, N]$$

$$r_{it} \geq 0 \quad \forall i \in [1, N]$$

Substituting the variable $g_{it} = \mu_{it} r_{it}$ $(\forall i \in [1, N])$, we get the following equivalent LP:

$$E^*(Q, \Lambda^t) = \min_G \sum_{i \in [1,N]} (g_{it} \frac{(e_{it} - e_i^{idle})}{\mu_{it}} + q_i e_i^{idle}) \quad (7)$$

subject to:

$$\sum_{i \in [1,N]} g_{it} \geq \lambda_{max}$$

$$g_{it} \leq \mu_{it} q_i \quad \forall i \in [1, N]$$

$$g_{it} \geq 0 \quad \forall i \in [1, N]$$

It can be shown that in this case the greedy solution to the LP problem, i.e., one in which the most energy efficient servers are allocated first, followed by the next most energy efficient servers and so on till the total service rate becomes at least equal to λ_{max}, is also the optimal solution. Note from Equation 7 servers will get picked in *decreasing* order of the coefficients $\frac{(e_{it} - e_i^{idle})}{\mu_{it}}$.

Greedy allocation of servers to the incoming job stream results in the power consumption being a piecewise linear function of the data center design vector Q, which is apparent from Equation 5. □

Lemma 2 allows us to re-write the Minimax constraint $E^*(Q, \Lambda^t) \leq \gamma$ as N linear constraints, i.e.,:

$$\left(\sum_{i=1}^{j-1} q_i e_{it}^{diff} + \frac{\lambda_{max} - \sum_{i=1}^{j-1} q_i \mu_{it}}{\mu_{jt}} e_{jt}^{diff} \right) \leq \gamma \quad \forall j \in [1, N]$$

Therefore, the robust heterogeneous data center design problem can be expressed as a LP with $N+1$ variables (data center design vector Q and γ) and $NM + 1$ constraints.

2.2 Illustrative Example

Wu use a simple example with two job types ($M = 2$) and two server types ($N = 2$) to illustrate the proposed approach. In this simple example, we assume that each server can process each job at the same rate of 1 job/second, i.e., $\mu_{ij} = 1$ $(i \in [1, 2]$ and $j \in [1, 2])$. Moreover, in terms of energy consumption, we assume that servers of type 1 are optimized to run jobs of type 1, while servers to type 2 are optimized to run jobs of type 2. In particular, $e_{11} = 1, e_{12} = 3, e_{21} = 4, e_{22} = 1$. The idle power consumption is assumed to be zero for both servers. Finally, each server is assumed to have unit cost and the maximum total arrival rate of jobs into the system is assumed to be 10 jobs/second. Figure 2 depicts the optimal data center design for different values of c_{budget}, which in this case simply constrains the total number of servers allowed in the data center. As can be seen from the figure, if only ten servers are allowed, the optimal design has 6 servers of type 1 and 4 servers of type 2, and dissipated 22 units of power in the worst case.

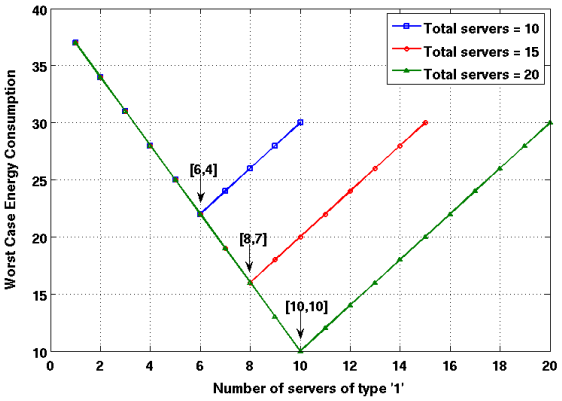

Figure 2: Illustrative Example

3. FUTURE WORK

The robust heterogeneous data center design problem we address in this paper is based implicitly on a number of assumptions that we are now working on relaxing. First, we assume that the data center performance specification only requires queue stability but does not account for either queuing or execution latency. We assumed also that the servers exist in one of two states, either ON or IDLE, but in general, servers have access to a range of power states which they can switch between at run-time. In addition, worst-case optimization using a minimax objective function is perhaps too pessimistic. We are currently looking at addressing scenario when some of the moments of arrival rate vector Λ are known. Much of this work falls squarely within the framework of adjustable robust optimization [1], with the data center design vector Q corresponding to the "'here-and-now"' variables, R (or any run-time scheduling decision) corresponding to the "'wait-and-see"' variables and Λ corresponding to the parametric uncertainty.

4. REFERENCES

[1] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 2004.

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

[3] V. Janapa Reddi, B. C. Lee, T. Chilimbi, and K. Vaid. Web search using mobile cores: quantifying and mitigating the price of efficiency. In *Proceedings of ISCA*, 2010.

[4] B. Lee and D. Brooks. Illustrative design space studies with microarchitectural regression models. In *IEEE HPCA*, 2007.

[5] R. Nathuji, C. Isci, and E. Gorbatov. Exploiting platform heterogeneity for power efficient data centers. In *Proceedings of IFAC*, 2007.