

# A Queuing Analysis of an Energy-Saving Mechanism in Data Centers

Christian Schwartz, Rastin Pries, Phuoc Tran-Gia

University of Würzburg, Institute of Computer Science, Würzburg, Germany.

Email: {christian.schwartz,pries,trangia}@informatik.uni-wuerzburg.de

**Abstract**—The high energy costs for running a data center led to a rethinking towards an energy-efficient operation of a data center. Designed for supporting the expected peak traffic load, the goal of the data center provider such as Amazon or Google is now to dynamically adapt the number of offered resources according to the current traffic load. In this paper, we present a queuing theoretical model to evaluate the trade-off between waiting time and power consumption if only a subset of servers is active all the time and the remaining servers are enabled on demand. We develop a queuing model with thresholds to turn-on reserve servers when needed. Furthermore, the resulting system behavior under varying parameters and requirements for Pareto optimality are studied.

## I. INTRODUCTION

In recent years, vast investments have been made to build up large-scale data centers. These data centers can have a size of a few hundred servers to up to 100,000 servers operating in a cloud, which is e.g. used for storage and computation. The reason for investments in those large clouds is the economy of scale for power, cooling, network, and administration capacity compared to classical enterprise systems. Although the data centers are set up for different purposes and applications, the basic structure is similar. The service provider aims to achieve high revenue while still guaranteeing the Service Level Agreements (SLAs). To decrease the capital expenditure, service providers - like Google - use commercial of-the-shelf hardware and to guarantee the SLAs, the data centers are designed according to the expected peak traffic load. However, the average load level of a data center is about 60% of the peak load [1]. Taking a look at the power consumption, these low loaded data centers waste a lot of energy. Although several servers are not under load, they still consume about 65% of the maximal power consumption [2].

The costs for powering the servers and network equipment exceed the acquisition costs after only three years. One way to efficiently reduce the power consumption in a data center is thus to switch off servers which are temporarily unused. In case the load in the data center again increases, the servers can be switched on using wake on LAN. The challenge is here to save as much energy as possible while still guaranteeing the SLAs.

In this paper, we develop a basic queuing model for mechanisms to operate a data center in an energy-efficient way. We separate the amount of servers in a data center in two groups: the *base-line servers* which are always in operation

and the *reserved servers* which can be switched on if needed. The decision to switch the server on or off are based on a hysteresis-oriented mechanism with two thresholds, which have to be properly dimensioned, depending on the load level expressed by the queue length of jobs waiting.

The remainder of this work is organized as follows. In Section II, the work related to energy-efficient data center operation is reviewed. Section III describes the considered dimensioning problems. The queuing model is outlined in Section IV and a closed form solution is presented in Section V. Results showing trade-offs between waiting time as performance measure and power consumption are evaluated in Section VI. Finally, Section VII concludes this paper.

## II. RELATED WORK

Several papers have been published, proposing new architectures for data centers, which provide more resilience or are cheaper to deploy [3]–[5]. However, only a few papers consider the energy consumption of a data center and propose mechanisms of how to reduce it.

Heller et al. [6] published a paper considering the trade-off between energy efficiency and resilience. They use the fat-tree architecture similar to [3], [4], which is based on commercial of-the-shelf network equipment. During normal network operation, the additional switches used for backup paths are switched off and only turned on in case of high load or network failures. The proposed mechanism is implemented in a testbed where OpenFlow is used for the switch management. However, they only turn off the switches and not the servers. Pries et al. [7] show that these only consume between 5% and 10% of the overall energy consumption.

Kliazovich et al. [8] developed a simulation environment for computing the energy consumption of different data center architectures. In addition to showing the share of network and server energy consumption, they present how much energy can be saved while using dynamic voltage and frequency scaling or dynamic power management.

One of the first paper presenting a dynamic resource management according to the offered load is presented by Chase et al. [9]. They propose an architecture where server clusters are dynamically resized in accordance to the negotiated SLAs.

A more detailed approach is presented by Chen et al. [10]. Three solutions are proposed to reduce the power consumption of servers in a data center. For the first solution, the workload

behavior of the near future is predicted while the second solution is a reactive solution, using periodic feedback of system execution. The third proposed solution is a hybrid solution using a combination of prediction and periodic feedback.

The most closely related work is presented in [11]–[13]. Their goal is to run a minimum number of servers in a data center to maximize the revenue of the service provider. The considered data center hosts a webpage application. While in [11], the authors do not consider user impatience and the fact that servers consume energy without producing revenue during wake up, [13] takes both into account. In [12], the authors introduced a policy for dynamically adapting the number of running servers. The goal of the paper was to find the best trade-off between consumed power and service quality.

Provisioning schemes for data centers are discussed in [14]. They discuss single and multi server models, where both the inter-arrival time as well as the service time are exponentially distributed and constant setup times are considered. Furthermore, a time-varying arrival process is discussed. With regard to optimality, *quasi*-optimality is proven for the discussed power-saving policies with regard to a compound objective function of mean service time and mean power consumption.

In [15] the authors present a model for server farms using exponential inter-arrival, service and setup times. They consider different policies for powering down servers for finite and infinite servers.

In contrast to the above mentioned papers, we evaluate the energy savings by adapting the number of servers dynamically according to the current load using thresholds on the queue size to enable additional servers and on the total number of jobs in the system to disable servers. Furthermore, we use a true multi-objective optimization approach to find optima, which can not be found using weighted sum aggregate objective functions.

### III. PROBLEM FORMULATION

A widely used data center architecture is the three-tier architecture shown in Figure 1. The upper two layers of the architecture are responsible for distributing the traffic and consist of layer 3 switches where each switch has a backup switch. In this paper, we focus on the edge layer and here on a single *Performance Optimized Data center* (POD). A POD consists of a number of servers connected over top of rack switches to an aggregation switch. We assume, that new jobs entering the system arrive with exponentially distributed inter-arrival time. When a job in form of a packet arrives at the POD, it is forwarded to an idle server. If no idle server is available, the job is queued. Once a server finishes processing its current job, it picks another one from the queue.

Our goal is now to evaluate how much power is consumed in a data center and how much can be saved when servers, not processing any job, are switched off. Therefore, we developed two different data center models. The first model, the *default data center*, consists of two-state servers only which are either *busy* or *idle* (see Figure 2a). For the second model,

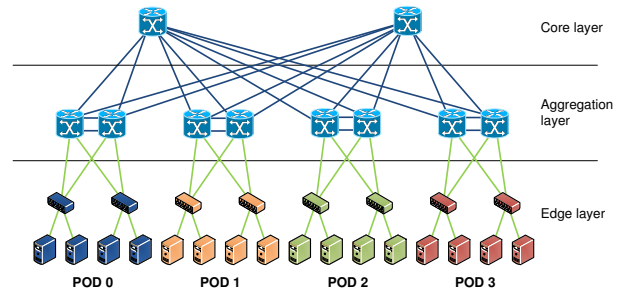
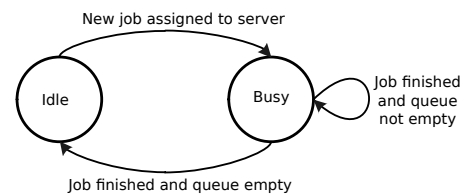
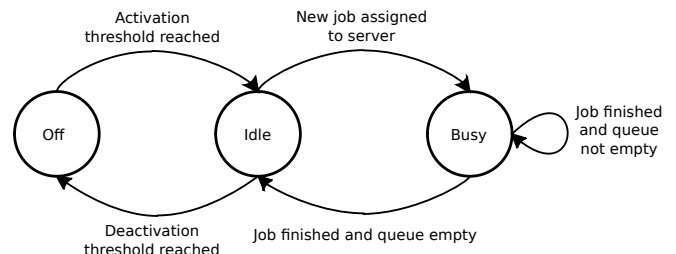


Fig. 1: Three-tier data center architecture.

a more *energy-efficient data center*, a subset of servers may additionally be switched on and *off* on demand (see Figure 2b) as recommended in [16].



(a) 2-state server model.



(b) 3-state model of a reserved server.

Fig. 2: Power state transition on a per server level.

#### A. Default Data Center

For the default data center model, each of the  $n$  servers is either on and processing a job or on and idle as depicted in Figure 2a. If a busy server finishes processing a job and the queue is empty, the server becomes idle. Once a new job is assigned to a yet idle server, the server becomes busy. According to our measurements of a server with an Intel twelve core processor (2.67 GHz) and 32 GB RAM, a server currently processing a job consumes  $e_{\text{busy}} = 240$  Watt. An idle server still consumes  $e_{\text{idle}} = 170$  Watt.

#### B. Energy-Efficient Data Center

For the second model, we differentiate between two types of servers. The number of base-line servers which are always on is given by  $n$  and the reserved servers to be enabled on demand is described by  $m$ . If they are enabled, the power consumption is similar to the default data center model. If they are disabled, each server consumes  $e_{\text{off}} = 0$  Watt. The  $n$  servers which are always on consume the same power as in the

default data center model. If the system queue has a length exceeding  $\theta_2$  (for  $\theta_2 \in (0, m)$ ), the  $m$  reserved servers are enabled and stay enabled until the total number of jobs in the system drops to  $\theta_1$  (for  $\theta_1 \in (0, n)$ ). The transition between power levels for each of the reserved servers is depicted in Figure 2b.

The energy-efficient data center operation model with the parameters  $\theta_1$  and  $\theta_2$  can be seen in Figure 3 and is described in detail in the next section.

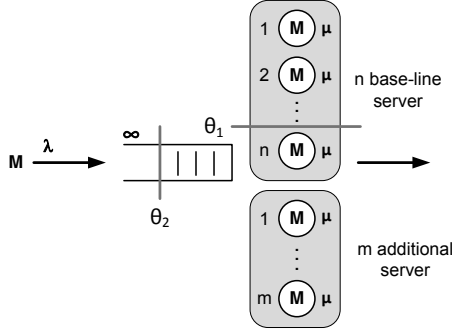


Fig. 3: System model for an energy-efficient operation.

#### IV. MODELING

In this section, we first discuss the default data center model, where a server can either be idle or busy, processing a job. Afterwards, the energy-efficient data center model with the three states is set up.

##### A. Default Data Center

Let us assume that the jobs in a POD arrive according to an independent Poisson process with rate  $\lambda$  and each server accepts only one job at a time with an exponentially distributed service time with mean  $\frac{1}{\mu}$ . Then, the system can be modeled using a simple  $M/M/n$  delay system. Where the random variable  $X$  gives the number of jobs in the system and  $x(i)$  is the stationary probability that  $i$  jobs are in the system.

We obtain the mean power consumption of such a system based on the measured values presented in Section III. If less than  $i < n$  jobs are in the system, then  $i$  servers are busy each consuming  $e_{\text{busy}}$  Watt and  $n - i$  servers are idle, where each consumes  $e_{\text{idle}}$  Watt. If  $i \geq n$  jobs are in the system all servers are busy and consume  $n \cdot e_{\text{busy}}$  Watt. This results in the upper bound for power consumption

$$E_{\text{max}} = \sum_{i=0}^n x(i)(ie_{\text{busy}} + (n-i)e_{\text{idle}}) + ne_{\text{busy}} \sum_{i=n+1}^{+\infty} x(i). \quad (1)$$

Furthermore, we can provide a lower bound for the power consumption of the system by assuming that a server is turned off if it is not processing a job, thus consuming  $e_{\text{off}}$ . By substituting  $e_{\text{off}}$  for  $e_{\text{idle}}$  in Equation 1 we get

$$E_{\text{min}} = \sum_{i=0}^n x(i)(ie_{\text{busy}} + (n-i)e_{\text{off}}) + ne_{\text{busy}} \sum_{i=n+1}^{+\infty} x(i).$$

##### B. Energy-Efficient Data Center

To extend the queuing system to model the energy-efficient data center model introduced in Section III-B, we need to modify the state space. We now model the system state as a tuple  $(i, j)$  where  $i$  is the number of jobs in the system and  $j$  is 1 if the reserved servers are activated and 0 if they are not activated (see Figure 4). The system activates the reserved servers if more than  $\theta_2$  jobs are in the queue, i.e. more than  $n + \theta_2$  jobs are in the system. The reserved servers are deactivated if the number of jobs in the system drops under  $\theta_1$ .

Again,  $X$  is the random variable describing the number of jobs in the system if the reserved servers are activated or deactivated, and  $x(i, j)$  is the stationary probability that  $i$  jobs are in the system, and the reserved servers are activated (for  $j = 1$ ) or deactivated (for  $j = 0$ ).

Based on the state space and transitions, we can formulate macro state equations, defined as the sum of all local balance equations of the states contained in the macrostate. They provide, when solved, the state probabilities required for further analysis.

First, we consider the macro state equations for state  $S_1^i$ , which contains all system states where up to  $i - 1$  jobs are in the system and no reserved servers are activated. Depending on  $i$ , we get the following equations.

$$\begin{aligned} i\mu x(i, 0) &= \lambda x(i - 1, 0) & 0 < i < \theta_1 \\ i\mu x(i, 0) + \theta_1 \mu x(\theta_1, 1) &= \lambda x(i - 1, 0) & \theta_1 \leq i \leq n \\ n\mu x(i, 0) + \theta_1 \mu x(\theta_1, 1) &= \lambda x(i - 1, 0) & n \leq i \leq n + \theta_2. \end{aligned}$$

Next, we examine the system state if the reserved servers are activated. The macro state  $S_2^i$  contains all system states with activated reserved servers and at least  $i + 1$  jobs in the system. We get

$$\begin{aligned} i\mu x(i, 1) &= \lambda x(i - 1, 1) \\ &+ \lambda x(n + \theta_2, 0) & \theta_1 < i \leq n + \theta_2 + 1 \\ i\mu x(i, 1) &= \lambda x(i - 1, 1) & n + \theta_2 + 1 < i \leq n + m \\ (n + m)\mu x(i, 1) &= \lambda x(i - 1, 1) & n + m < i. \end{aligned} \quad (2)$$

The third macro state  $S_3$  contains all system states where only the base-line servers are activated and its equation states

$$\lambda x(n + \theta_2, 0) = \theta_1 \mu x(\theta_1, 1).$$

Finally, the sum of all probabilities should be 1, i.e.

$$1 = \sum_{i=0}^{n+\theta_2} x(i, 0) + \sum_{i=\theta_1}^{+\infty} x(i, 1)$$

should hold.

Based on the state probabilities, we can derive the required performance metrics for our analysis as we did in Section IV-A.

The carried traffic and utilization is given by

$$a = \frac{\lambda}{\mu} \quad \text{and} \quad \rho = \frac{\lambda}{\mu(n+m)}.$$

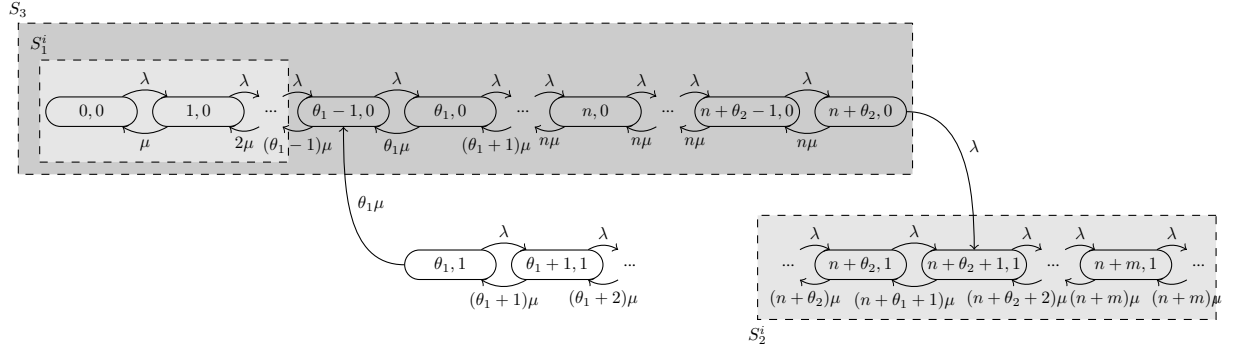


Fig. 4:  $M/M/(n+m)^{(\theta_1, \theta_2)}$  System with macro states  $S_1^i$ ,  $S_2^i$ , and  $S_3$  for the calculation of  $x(i, \{0, 1\})$ .

Furthermore, we obtain the mean queue length

$$\Omega = \sum_{i=n}^{n+\theta_2} (i-n)x(i,0) + \sum_{i=n+m}^{+\infty} (i-(n+m))x(i,1).$$

By applying macro state Equation 2 we obtain for all  $i > n+m$

$$x(i,1) = \rho x(i-1,1) = x(n+m,1)\rho^{i-(n+m)}.$$

Using this result and the first derivative of the geometric series we get

$$\begin{aligned} \Omega &= \sum_{i=n}^{n+\theta_2} (i-n)x(i,0) + x(n+m,1) \sum_{i=0}^{+\infty} i\rho^i \\ &= \sum_{i=n}^{n+\theta_2} (i-n)x(i,0) + x(n+m,1) \frac{\rho}{(1-\rho)^2}. \end{aligned}$$

Now, we can give the mean waiting time for all jobs in the system as

$$E[W] = \frac{\Omega}{\lambda}.$$

Finally, we obtain the mean power consumption similarly to Equation 1.

$$\begin{aligned} E &= \sum_{i=0}^n x(i,0)(ie_{\text{busy}} + (n-i)e_{\text{idle}} + me_{\text{off}}) \\ &+ \sum_{i=n+1}^{n+\theta_2} x(i,0)(ne_{\text{busy}} + me_{\text{off}}) \\ &+ \sum_{i=\theta_1}^{n+m} x(i,1)(ie_{\text{busy}} + (n+m-i)e_{\text{idle}}) \\ &+ x(i > n+m)(n+m)e_{\text{busy}} \end{aligned}$$

## V. CLOSED FORM SOLUTION

We obtain closed form solutions for the state probabilities. These equations can be derived by recursively applying the macro state equations. All equations feature a factor  $x(0,0)$ , which in turn can be calculated using the normalization property. Due to the length of the individual formulas, the following shorthand is introduced: For each state probability  $x(i,j)$  depending on the factor  $x(0,0)$  we define  $\bar{x}(i,j) = \frac{x(i,j)}{x(0,0)}$

(i.e. we cancel the factor).

For  $0 < i < \theta_1$  we get

$$x(i,0) = x(0,0) \cdot \frac{a^i}{i!}.$$

As a further shorthand for substitution, we define

$$s_i = \sum_{k=0}^i a^k (n-k-1)!.$$

Using this definition, we get the state probability for  $\theta_1$  jobs in the system with activated reserved servers:

$$\begin{aligned} x(\theta_1,1) &= x(0,0) \cdot \frac{a^{n+\theta_2+1}}{\left(1 + \frac{a}{\theta_1}\right)} \\ &\cdot \frac{(\theta_1-1)! \left(\frac{(1-a^{\theta_2})\theta_1}{1-a} + a^{\theta_2} s_{n-\theta_1}\right)}{n^{\theta_2} n! (n-\theta_1+1)!}. \end{aligned}$$

For  $\theta_1 \leq i \leq n$  we get

$$\begin{aligned} x(i,0) &= x(0,0) \cdot \left( \frac{\bar{x}(n,0)a^{i-\theta_1+1}}{(i-\theta_1+1)!} \right. \\ &\quad \left. - \frac{\bar{x}(\theta_1,1)\theta_1 s_{i-\theta_1}}{i!} \right). \end{aligned}$$

And for  $n < i \leq n+\theta_2$  we get

$$\begin{aligned} x(i,0) &= x(0,0) \cdot \left( \frac{\bar{x}(n,0)a^{i-\theta_1+1}}{n^{i-n}(n-\theta_1+1)!} \right. \\ &\quad \left. - \bar{x}(\theta_1,1) \left( \frac{\theta_1 s_{n-\theta_1} a^{i-n}}{n!} \right. \right. \\ &\quad \left. \left. + \frac{\theta_1(1-a^{i-n})}{1-a} \right) \right). \end{aligned}$$

Thus, we have all probabilities for system states where only the baseline servers are active. For the reserved servers, we obtain state probabilities for  $\theta_1 < i \leq n+\theta_2+1$  as

$$\begin{aligned} x(i,1) &= x(0,0) \cdot \left( \bar{x}(\theta_1,1) \frac{a^{i-\theta_1}\theta_1!}{i!} \right. \\ &\quad \left. + \bar{x}(n+\theta_2,0) \sum_{k=1}^{i-\theta_1} \frac{a^k(i-k)!}{i!} \right). \end{aligned}$$

For  $n + \theta_2 + 1 < i \leq n + m$  we get

$$x(i, 1) = x(0, 0) \cdot \bar{x}(n + \theta_2 + 1, 1) \cdot \frac{a^{i-(n+\theta_2+1)}(n + \theta_2 + 1)!}{i!},$$

and finally for  $i > n + m$

$$x(i, 1) = x(0, 0) \cdot \bar{x}(n + m, 1) \left( \frac{a}{n + m} \right)^{i-(n+m)}.$$

As discussed earlier, the probability of an empty system is given by the usual normalization property:

$$x(0, 0) = \left( 1 + \sum_{k=1}^{n+\theta_2} \bar{x}(k, 0) + \sum_{k=\theta_1}^{\infty} \bar{x}(k, 1) \right)^{-1}.$$

Using these equations, the state probabilities can be obtained and the performance metrics introduced in Section IV can be evaluated.

## VI. PERFORMANCE EVALUATION

Based on the metrics obtained in Section IV, we can now compare the introduced default data center and energy-efficient data center models.

An optimal system setting would decrease both, waiting time and power consumption. For the discussion of this optimization problem, we require additional notation which is introduced first. We assume that the job inter-arrival rate  $\lambda$ , the job service rate  $\mu$ , and the total number of servers  $n_{\text{total}}$  are constants and not subject to the optimization process. Thus, the complete system can be described by the number of base-line servers  $n$  from which the number of reserved servers  $m$  can be easily derived if the total number of servers  $n_{\text{total}}$  is known, the server activation threshold  $\theta_2$ , and the server deactivation threshold  $\theta_1$ . Given these parameters, we define  $e(n, \theta_1, \theta_2)$  to be the mean power consumption of the system and  $w(n, \theta_1, \theta_2)$  to be the mean waiting time of all jobs.

A general approach for solving such multi objective optimization problems is defining a single aggregate objective function, such as:

$$f(n, \theta_1, \theta_2) = \alpha e(n, \theta_1, \theta_2) + (1 - \alpha)w(n, \theta_1, \theta_2) \quad (3)$$

for  $0 \leq \alpha \leq 1$ . Then, it is possible to choose an  $\alpha$  in such a way that a desirable trade-off is made. Thus, the optimization problem can be defined as

$$\begin{aligned} \min f(n, \theta_1, \theta_2) \quad s.t. \quad & 1 < n < s, \\ & 1 < \theta_1 < n - 1, \\ & 1 < \theta_2 < m - 1, \end{aligned} \quad (4)$$

and trivially solved by evaluating all valid parameter combinations, sorting the objective function values and choosing the minimum.

This approach has the obvious disadvantage that while a parameter combination may be optimal according to the chosen objective function, it may very well not be optimal to the user. For example, it may be possible that another system

configuration exists with a minimally greater mean waiting time and a greatly reduced power consumption. To be able to decide whether such a trade-off exists, a more global view of the problem space is required. However, due to the number of possible parameter combinations, it is difficult to select appropriate parameters. We reduce the number of possible parameters by considering only Pareto optimal states.

To define Pareto optimality, we need to introduce the product order partial relation. Let  $X \subseteq \mathbb{R}^n$  be our feature set. We set  $x \preceq x^*$  for  $x, x^* \in \mathbb{R}^n$  iff

$$x_i \leq x_i^* \quad \forall 1 \leq i \leq n \quad (5)$$

holds. Then,  $x^*$  is Pareto optimal in  $X$  if no  $x \in X \setminus \{x^*\}$  exists, such that  $x \preceq x^*$  holds.

To study the system behavior, we consider an exemplary rack of  $n_{\text{total}} = 100$  servers, where new jobs arrive with a negative exponential inter-arrival time with mean 10 ms, yielding  $\lambda = \frac{1}{10}$  ms. To determine the mean service time we turn to [17] where it is reported that on average servers are operating at 10 to 50% of their maximum utilization levels. With this in mind, we assume that the service time for job completion is again negative exponential with a mean of 400 ms, which implies  $\mu = \frac{1}{400}$  ms, resulting in an overall utilization of  $\frac{\lambda}{\mu(n+m)} = 0.4$ , well within the described limits.

Based on these parameters, we can compute the mean waiting time and power consumption for the default data center model. The mean waiting time achieved by the default data center model provides a lower bound for the achievable waiting time for the energy-efficient data center model, as obviously all  $n$  servers are always either idle or busy. For the parameters described above, the default data center model achieves a mean waiting time for all jobs  $E[W] = 4.75 \cdot 10^{-14}$  ms. Furthermore, the mean power consumption of the system, under the assumption that no servers are disabled, is set at  $E_{\text{max}} = 100\%$ , which provides an upper bound for the energy-efficient data center model. However, if we assume that all servers are immediately switched off if they are not processing any jobs, we get  $E_{\text{min}} = 48.48\%$ , which is the lower bound for the energy-efficient data center model.

Using the same parameters, we evaluate the systems performance metrics, the mean power consumption  $e(n, \theta_1, \theta_2)$  relative to  $E_{\text{max}}$ , and the mean waiting time  $w(n, \theta_1, \theta_2)$  for the energy-efficient data center. As mentioned before, the Pareto optima of the system are a subset of the  $\mathbb{R}^2$ , with one dimension corresponding to the mean waiting time, the other to the mean power consumption. We plot all Pareto optima, cf. Figure 5, the resulting curve has hyperbolic properties, going asymptotically to the mean waiting time as well as asymptotically to a parallel of the lower bound of the power consumption. This allows us to select an acceptable increase in waiting time, for example one that would still satisfy a service level agreement, and harness the resulting energy savings. On the other hand, we can decide on the required energy savings and infer if the corresponding waiting times are acceptable. One possible parameter choice would allow the reduction of

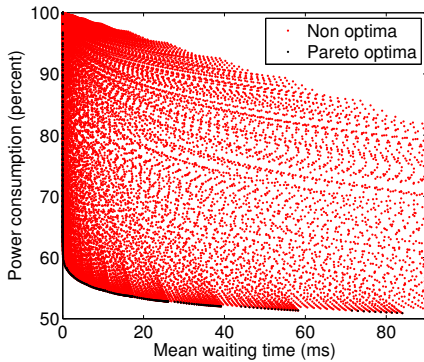


Fig. 5: Set of Pareto optimal values.

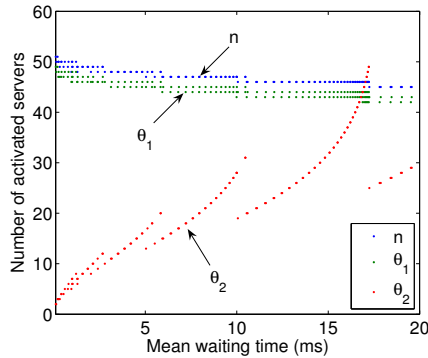


Fig. 6: Set of Pareto optimal values.

the energy consumption by 40% while only increasing the waiting time by less than one millisecond.

Given the set of Pareto optima, we can investigate the parameter choice that leads to these optima. To this end, we plot the system parameters for each optimum, cf. Figure 6. The optima themselves are sorted according to the mean waiting time. From the figure we can see that generally, the server deactivation threshold is very close to the number of base-line servers, in most cases  $n - \theta_1 = 2$ , the closest possible distance due to the macro state equation constraints. Furthermore, the number of base-line servers is decreasing as the waiting time increases. The mean waiting time spectrum can be partitioned in interleaving sections, during which the number of base-line servers remain constant. Furthermore, in such a section, the server activation threshold increases super linearly.

## VII. CONCLUSION

In this paper, we proposed a threshold-oriented operation model for reducing the power consumption in a data center. This is achieved by adjusting the number of active servers in the system while turning the others off. The decision of whether servers should be activated is based on the number of packets in the incoming queue of the top of rack switch. To get the optimal trade-off between average waiting time in the system and the overall power consumption, we set up an analytical model.

The results show that configurations exist, so that the power consumption can be significantly reduced while still having an acceptable mean waiting time. Thus, we can guarantee the service level agreements to the end user while still saving about 40% of energy with the server adaptation. Possible directions for future research can introduce more adjustment steps, take the time required for waking servers up into account, and study the dynamic behavior of the system.

## REFERENCES

- [1] T. Benson, A. Akella, and D. Maltz, "Network traffic characteristics of data centers in the wild," in *Internet Measurement Conference (IMC)*, Melbourne, Australia, November 2010.
- [2] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 89–73, 2009.
- [3] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *SIGCOMM '08: Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, Seattle, WA, USA, August 2008, pp. 63–74.
- [4] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 51–62, 2009.
- [5] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "BCube: a high performance, server-centric network architecture for modular data centers," in *SIGCOMM '09: Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, Barcelona, Spain, August 2009, pp. 63–74.
- [6] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastic tree: Saving energy in data center networks," in *7th USENIX Symposium on Networked System Design and Implementation (NSDI)*, San Jose, CA, USA, April 2010, pp. 249–264.
- [7] R. Pries, M. Jarschel, D. Schlosser, M. Klopff, and P. Tran-Gia, "Power Consumption Analysis of Data Center Architectures," in *GreenNets*, Colmar, France, Oct. 2011.
- [8] D. Kliazovich, P. Bounvry, Y. Audzevich, and S. U. Khan, "Greencloud: A packet-level simulator of energy-aware cloud computing data centers," in *IEEE Globecom*, Miami, FL, USA, December 2010.
- [9] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing energy and server resources in hosting centers," in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, Chateau Lake Louise, Banff, Canada, October 2001.
- [10] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautham, "Managing server energy and operational costs in hosting centers," *ACM SIGMETRICS Performance Evaluation Review*, vol. 33, no. 1, pp. 303–314, 2005.
- [11] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing cloud providers revenues via energy aware allocation policies," in *IEEE International Conference on Cloud Computing*, Miami, Florida, July 2010, pp. 131–138.
- [12] D. Dyachuk and M. Mazzucco, "On allocation policies for power and performance," in *11th ACM/IEEE International Conference on Grid Computing (Grid 2010) – Energy Efficient Grids, Clouds and Clusters Workshop (E2GC2-2010)*, Brussels, Belgium, October 2010.
- [13] M. Mazzucco, D. Dyachuk, and M. Dikaiakos, "Profit-aware server allocation for green internet services," in *18th Annual IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer Systems*. Miami, Florida: IEEE Computer Society, August 2010, pp. 277–284.
- [14] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, 2010.
- [15] A. Gandhi, M. Harchol-Balter, and I. Adan, "Server farms with setup costs," *Performance Evaluation*, 2010.
- [16] United States Environmental Protection Agency, "Data center report to congress," <http://1.usa.gov/IEAPHC>, 2007.
- [17] L. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.