Task Scheduling and Server Provisioning for Energy-Efficient Cloud-Computing Data Centers

Ning Liu
Department of Mathematics
New Jersey Institute of Technology
Newark, NJ 07102
nl59@njit.edu

Ziqian Dong
Department of Electrical and Computer Engineering
New York Institute of Technology
New York, NY 10023
ziqian.dong@nyit.edu

Roberto Rojas-Cessa
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ 07102
rojas@njit.edu

***Abstract*—In this paper, we present an optimization model for task scheduling for minimizing energy consumption in cloud-computing data centers. The proposed approach was formulated as an integer programming problem to minimize the cloud-computing data center energy consumption by scheduling tasks to a minimum numbers of servers while keeping the task response time constraints. We prove that the average task response time and the number of active servers needed to meet such time constraints are bounded through the use of a greedy task-scheduling scheme. In addition, we propose the most-efficient-server-first task-scheduling scheme to minimize energy expenditure as a practical scheduling scheme. We model and simulate the proposed scheduling scheme for a data center with heterogeneous tasks. The simulation results show that the proposed task-scheduling scheme reduces server energy consumption on average over 70 times when compared to the energy consumed under a (not-optimized) random-based task-scheduling scheme. We show that energy savings are achieved by minimizing the allocated number of servers.**

***Index Terms*—Cloud computing, Energy, Green data centers, Task Scheduling, Greedy Algorithm, Integer Programming.**

I. Introduction

Cloud computing has risen as a new computing paradigm to bring unparalleled flexibility and access to shared and scalable computing resources [1]–[4]. Cloud services are usually implemented in one or multiple data centers where a large number of servers, storage units, and a telecommunication infrastructure are provisioned. As these data centers grow from hundreds to hundreds of thousands of servers to meet the increasing demand, the energy cost of these large-scale data centers contributes to a major portion of the operating costs [5]–[7].

The major energy-consumption components of a datacenter are servers, interconnecting telecommunication networks, and cooling systems [8]. The interconnecting network is composed of a large number of switches/routers that enable the communication among servers of data centers [9]–[13]. The energy consumed by the interconnect has been reduced by strategically turning off equipment that is not being used [14], [15]. However, an inefficient use of servers results in significant energy consumption that may minimize the savings obtained from an efficient interconnect [7], [16].

A lot of research has been done to study different aspects on consumption and management of the datacenter energy and allocation of datacenter computing resources [17]–[24]. A desirable approach to reduce energy consumption is to reduce the number of active servers (an active server is one in ON state and ready to receive and process tasks). However, due to the slow boot-up process of servers, the number of servers is over-provisioned in cloud-computing data centers to meet sudden increase of service demand. A dynamic resource allocation based on workload scheme that incorporates network traffic management and server workload studies the tradeoff of server workload consolidation with respect to performance and reliability. This study shows energy savings of up to 75% with the proposed traffic management and server workload consolidation scheme [20].

Schemes targeting a reduction of energy consumption by cooling systems focus on the study of thermal features of power distribution in large-scale data centers. A study on indirect detection of power dissipation, such as temperature distribution in the data center, was aimed at reducing data center cooling costs [21]. These schemes work as closed-loop systems where the sensed temperature is used to determine the cool zones to where tasks can be assigned. A thermal and power-aware task scheduling for hadoop-based storage centric data centers was proposed to ensure the servers in the data center operate at a temperature below a certain threshold to reduce the power needed by the cooling system [22].

Other approaches focus on financial aspects of operating data centers. A utility function using pricing models for real-time electricity cost was recently proposed [24]. This method exploits the pricing difference of electricity at different locations and times to control the load of the data centers at different locations for optimized profits. These approaches, favored by the service providers, do not solve the fundamental problem of reducing cloud-computing data center energy consumption.

These preliminary works indicate that a large portion of the energy consumption of a cloud-computing data center is based on the number of active servers and their computational loads. It also demonstrates the potential of reducing energy consumption by increasing data center resource utilization.

In this paper, we model the data center energy consumption with respect to task response time and the number of servers required to meet the expected task processing deadlines as an integer programming optimization problem. We show that the number of servers is bounded to achieve the optimized energy consumption in a deadline-constrained task environment. In addition, we propose a greedy task-scheduling scheme for min-

imizing the energy consumption while observing constrained average task response times. The proposed most-efficient-server first scheme assigns tasks to the most energy-efficient servers first. We simulate the proposed scheme and evaluate its impact on energy savings of a data center, and compare it to those savings of a random-based task scheduling scheme. Our simulation results show that the proposed task-scheduling scheme can reduce server energy consumption by an average of 70 times of that consumed by the random-based task-scheduling scheme.

The remainder of this paper is organized as follows. Section II presents the proposed energy consumption model for a cloud-computing data center. Section III introduces the most-efficient-server first task-scheduling scheme. Section IV presents simulation results of energy consumption and task response time for the proposed and random-based task-scheduling schemes. Section V presents our conclusions.

II. Task scheduling and energy consumption

The cloud-computing data center houses hundreds of thousands of servers and storage units interconnected through an interconnection network resembling a fat-tree topology, as shown in Figure 1. In this paper, we assume that the network infrastructure provides enough bandwidth to avoid delays caused by the network.

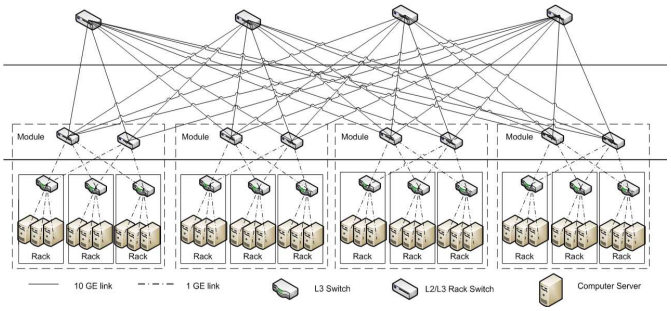

Fig. 1. Networked servers architecture inside the Cloud.

The operation of the cloud-computing data center is described as follows. Users send requests to the data center for computing jobs, named tasks in the remainder of this paper. A task may include entering data, processing, accessing software, or storage functions. The data center classifies tasks according to the service-level agreement and requested services. Each task is then assigned to one of the available servers. In turn, the servers perform the requested task, and a response, or result, is transmitted back to the user.

Energy consumption of data centers includes many factors, such as servers, load, interconnection network, cooling system, power distribution system, and etc. In this paper, we focus on the study of efficient task scheduling to minimize data center energy consumption by reducing the number of servers. We formulate the energy consumption of the data center with respect to the number of active servers as an integer-programming problem with the objective of minimizing data center energy consumption.

TABLE I
Terminology Definition

Terminology	Definition
N	Number of task types
n_i	Number of type-i task arrivals, where $0 \leq i \leq N$
M	Total number of servers in data center
S_j	Server j, where $1 \leq j \leq M$
$B_{i,j}$	Capacity of S_j to store type-i tasks
$x_{i,j}$	Number of task i assigned to S_j
$\mu_{i,j}$	Average processing time of type-i task by S_j
$\tau_{i,j}$	Average queuing delay of type-i tasks on S_j
$w_{i,j}$	Queue occupancy of type-i tasks at S_j
T_w	Average response time per task
X_j	Schedule in which tasks are processed by S_j
ω	Weight vector
$P_{i,j}$	Power consumed by S_j to process a type-i task
E	Total server energy consumption in a data center

A. Optimization of energy consumption with respect to minimizing the number of active servers

A data center is required to handle tasks requiring different computational resources. Therefore, servers may provide different task response times (assuming that a complete task is assigned to a single server) and consume different levels of energy for different types of tasks. Table I lists the definitions and nomenclature used in this paper. We consider N different types of tasks. The number of tasks of type i is n_i, where $1 \leq i \leq N$. The number of servers in a data center is M, and each server is denoted as S_j, where $1 \leq j \leq M$.

Tasks are associated with a deadline or the maximum allowable time that the data center may take to send a response to the task source. For simplification and without losing generality, we set the queue capacity of S_j for type i tasks equal to the deadline or $B_{i,j}$. $x_{i,j}$ denotes the number of type-i tasks assigned to S_j by a central scheduler (see Figure 2), $\mu_{i,j}$ denotes the processing time for a type-i task at S_j, $\tau_{i,j}$ denotes the average queuing delay of type-i task at S_j, T_w denotes the average response time per task, which is the sum of queuing delay and the task processing time, and $w_{i,j}$ denotes the number of type-i tasks queued at S_j at a given time. Figure 2 shows an example of the task scheduler and the servers. The centralized scheduler can also be implemented in a distributed manner [25]. X_j denotes the task schedule vector for S_j, and ω as the weight (column) vector:

$$\omega = \Big(\sum_{i=1}^{N} x_{i,j} - 1,\ \sum_{i=1}^{N} x_{i,j} - 2,\ \cdots, 1,\ 0 \Big)^T \qquad (1)$$

Here, X_j is defined as an $N \times \sum_{i=1}^{N} x_{i,j}$ matrix with elements of either 0 and 1, representing the scheduled sequence in which tasks are scheduled. The row of the matrix indicates the task type and the column indicates the time slot or sequence in which tasks are scheduled. For example, for three types of tasks ($N = 3$) and four tasks scheduled for S_j, and the schedule vector is presented as a 3x4 matrix,

$$X_j = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

where the top row shows two type-1 tasks, the second row

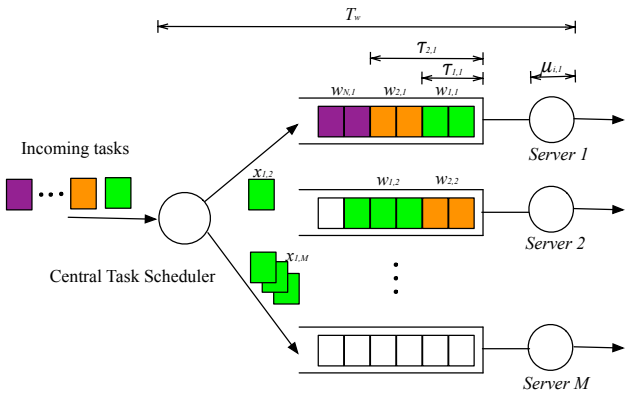

Fig. 2. Data center task scheduler model.

shows one type-2 task, and the bottom row shows one type-3 task. The tasks are scheduled in the order as the sequence shown by the columns.

To formulate the data center energy consumption, we denote the power consumed by S_j to complete a type-i task as P_{ij} and the amount of energy the data center servers consumes as E. The total data center server energy consumption is defined as the sum of energy consumed by servers processing all the tasks at the data center for a given period of time. The amount of power consumed by each task, the number of tasks, and the time to process the task define the total amount of consumed energy:

$$E = \sum_{i=1}^{N}\sum_{j=1}^{M} \mu_{i,j} P_{i,j} x_{i,j}. \tag{2}$$

We set the objective function as finding an optimum assignment of $x_{i,j}$ such that the time a server takes to complete the assigned tasks is minimized, which in turn minimizes E.

We formulate the optimization problem under the following two cases, 1) when the data center has not received any task initially, therefore the upcoming tasks are immediately assigned to the servers, and 2) when the data center has backlogged tasks in it, such that upcoming tasks have to be queued until they can be assigned for service.

1) No backlogged tasks: When the data center has no backlogged tasks and the upcoming ones are immediately assigned, the optimization problem is formulated as

$$\begin{aligned} \min_{x} \quad & E(x) = \sum_{i=1}^{N}\sum_{j=1}^{M} \mu_{i,j} P_{i,j} x_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{M} x_{i,j} = n_i,\ x_{i,j} \le B_{i,j} \end{aligned} \tag{3}$$

Then T_w is simply bounded by $T_j\ X_j\ \omega$, where

$$T_j = (\mu_{1,j},\ \mu_{2,j},\ \cdots,\ \mu_{N,j}) \tag{4}$$

2) With queuing delay: When the servers of the data center are busy and have queued tasks, upcoming tasks are queued if there is available room in the queues. In this case, the optimization problem is formulated as

$$\begin{aligned} \min_{x} \quad & E(x) = \sum_{i=1}^{N}\sum_{j=1}^{M} \mu_{i,j} P_{i,j} x_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{M} x_{i,j} = n_i,\ x_{i,j} \le B_{i,j} - w_{i,j} \end{aligned} \tag{5}$$

The average response time, composed of both processing and queuing delays, is calculated as

$$T_w = \frac{\sum_{j=1}^{M}\Big[\sum_{i=1}^{N} x_{i,j}\tau_{i,j} + T_j X_j \omega\Big]}{\sum_{i=1}^{N} n_i}. \tag{6}$$

B. Analysis of single task type

In the remainder of this section, we analyze the assignment of a single task type ($N = 1$) and give bound of the number of servers required. This can be considered under the assumption that other types can be decomposed into a linear combination of a unit type. For simplicity, we remove the subscript i in the notation for $\mu_{i,j}$, $x_{i,j}$,$\tau_{i,j}$ in the remainder of this section. Eqn. (6) is represented as:

$$T_w = \sum_{j=1}^{M} \frac{\mu_j}{2} x_j (x_j - 1)$$

with $\sum_{j=1}^{M} x_j = n$, where n is the number of tasks in the data center.

Let us assume that

$$F(x_j, \lambda) = T = \sum_{j=1}^{M} \frac{\mu_j}{2} x_j (x_j - 1) + \lambda(\sum_{j=1}^{M} x_j - n) \tag{7}$$

To find the minimum response time, we take the partial derivatives of F with respect to x_j and λ:

$$\frac{\partial F}{\partial x_j} = \frac{\mu_j}{2}(2x_j - 1) + \lambda = 0 \tag{8}$$

$$\frac{\partial F}{\partial \lambda} = \sum_{j=1}^{M} x_j - n = 0. \tag{9}$$

We then obtain

$$x_j = \frac{1}{2} - \frac{\lambda}{\mu_j},\ \text{for } j = 1,\ \ldots,\ M \tag{10}$$

$$\frac{M}{2} - \lambda \sum_{j=1}^{M} \frac{1}{\mu_j} = n \tag{11}$$

From (10) and (11), we get

$$\lambda = \frac{M - 2n}{2\sum_{j=1}^{M} \frac{1}{\mu_j}}, \tag{12}$$

$$x_j = \frac{1}{2} - \frac{M - 2n}{2\mu_j \sum_{j=1}^{M} \frac{1}{\mu_j}} \tag{13}$$

Since $1 \leq x_j \leq B_j$, the number of servers, M, is bounded by

$$2n - (2B_j - 1)\mu_j \sum_{j=1}^{M} \frac{1}{\mu_j} \leq M \leq 2n - \mu_j \sum_{j=1}^{M} \frac{1}{\mu_j} \tag{14}$$

Therefore, the range of n that can be allocated to the servers with the following bound defined by M, B_j, and μ_j, is defined by

$$\frac{1}{2}(M + \mu_j \sum_{j=1}^{M} \frac{1}{\mu_j}) \leq n \leq \frac{1}{2}(M + (2B_j - 1)\mu_j \sum_{j=1}^{M} \frac{1}{\mu_j}) \tag{15}$$

Given a distribution of n, we can provision the optimized number of servers required to process a given amount of tasks.

III. The most-efficient-server first scheme

Should servers with higher computing capacity be available, where the energy consumption is directly proportional to the server capacity (defined as the total number of tasks a server can process in parallel), these servers become the most preferred ones. In this case, the optimization problem can be interpreted as a greedy-assignment scheme. For this, it is considered that the central scheduler sorts the servers based on energy efficiency and assigns tasks to the most energy-efficient servers first and it then continues to allocate tasks to the second most efficient servers on the list, and so on, until no task remains or else, servers' queues are full.

We present a practical greedy scheme to achieve the assignment of tasks to the set of the most efficient servers first. For a data center with a single server type, the scheme aims to assign the larger number of task to an active server until the saturation point (where the server performance starts to decay) before starting assignments to a new and idle server. The greedy scheduling scheme is described according to the following pseudo-code:

Algorithm 1 Most-efficient-server first scheme.

Input: Set of tasks and servers
Output: Scheduling of tasks to servers
$a = b = c = 1$
for *Each Task x of type i* **do**
  for *Each Server j* **do**
    Calculate server energy consumption $E(i,j) = P_{i,j}\mu_{i,j}$
    if *E(i,j) $\leq$ E(a,b)* **then**
      $a = i$ $b = j$
    end
  end
end
Schedule(a to Server b)
while *unscheduled tasks (denoted as $y_{i'}$ of type i') remain* **do**
  for *Each Server j* **do**
    Calculate server energy consumption $E(y_{i'},j)$
    if *$E(y_{i'},j) \leq E(y_{i'},c)$* **then**
      $c = j$
    end
  end
  Schedule($y_{i'}$ to Server c)
end

The central task scheduler maintains a sorted list of available servers and their energy profiles with the most energy-efficient servers on the top of the list. Upon receiving task requests, it assigns tasks to the servers on top of the list. The servers receive task assignments and update their energy profile at the central scheduler. Once the most energy-efficient servers are saturated, the unscheduled tasks will be assigned to the less energy-efficient servers.

IV. Simulation Results

We modeled a data center with a central scheduler and a number of servers that handles heterogenous tasks using Matlab for evaluation of the energy consumption through computing simulation. The performance evaluation was based on the total and the average task response times for the proposed most-efficient-server first task-scheduling scheme. We limited the number of servers to 100. Note that a larger number of servers in the simulation may provide similar normalized results obtained in our experiments but at the expense of longer simulation time.

We compare the performance of the proposed scheduling scheme with a random-based task-scheduling scheme [26]. The random-based task-scheduling scheme assigns tasks to servers on a random basis without additional constraints for task allocation or server selection, except for considering available queue in each server. We simulated both schemes using 20 types of tasks ($N = 20$) and exponentially distributed task arrivals. We conducted 1000 experiments (one experiment is a task allocation trial with a duration of sufficient events to allow the distribution of the task to complete) for each scheduling scheme.

Figure 4 shows the histogram of the energy consumption of the performed experiments. We represent energy consumption in general energy units in the following figures. The energy savings achieved with the proposed scheduling scheme may vary as the servers' properties do. This histogram shows the

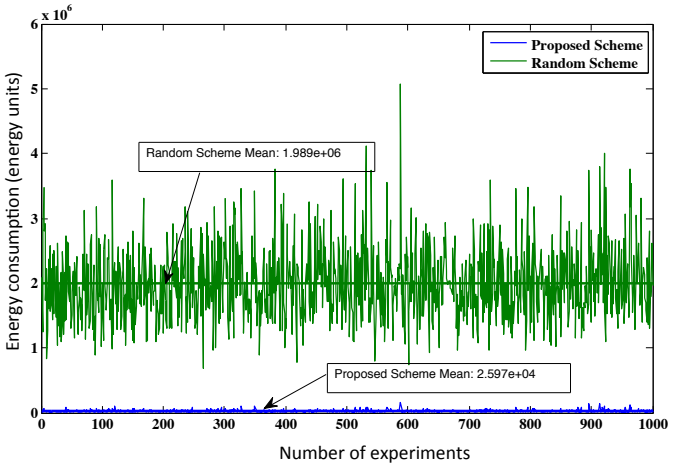

Fig. 3. Servers' energy consumption of the proposed and random task-scheduling schemes for all experiments.

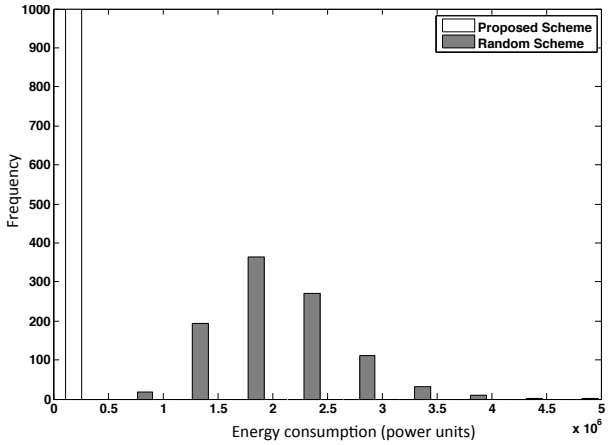

Fig. 4. Histogram of the total server energy consumption for the proposed and random task-scheduling schemes of a data center.

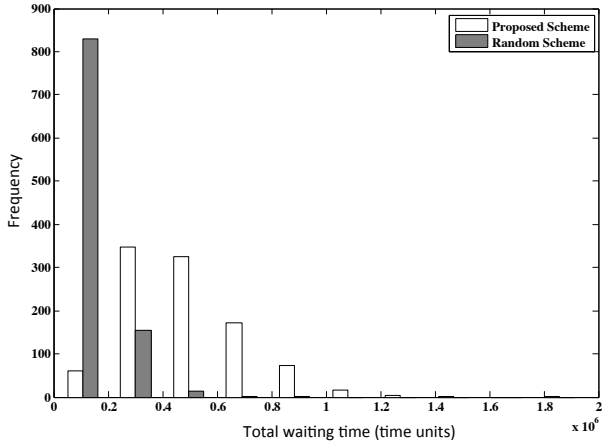

Fig. 5. Histogram of total task response time for the proposed and random scheduling schemes.

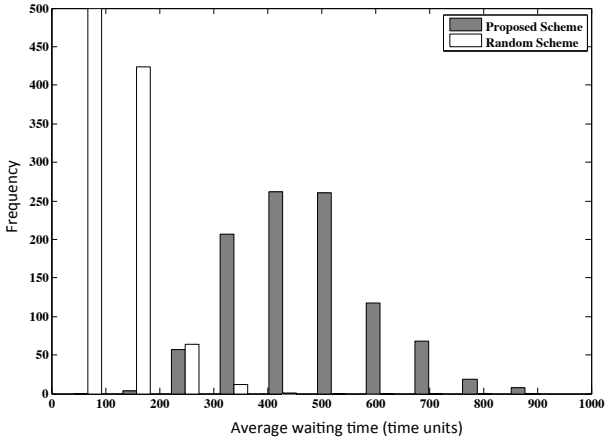

Fig. 6. Histogram of average task response time for the proposed and random scheduling schemes.

impact on energy consumption for different task-scheduling schemes. The proposed task-scheduling scheme consumes an average of over 70 times (by comparing the mean energy consumption of both schemes, $\frac{1.989e+06}{2.597e+04}$) less than the data center using the random-based task-scheduling scheme. Specifically, the distribution of the random-based scheme presents a distribution around the mean with larger deviations than that of the proposed scheme. This occurs as the proposed scheme achieves the highest efficiency as the servers assigned are those needed and the increment of number of servers that become active are proportional to the number of tasks. The random-based scheme may or may not re-use a server in an efficient manner.

Figure 5 shows the histogram of the total task response time for the proposed and random-based task-scheduling schemes. It shows that the response time distribution of the proposed scheme has a larger deviation than that observed for the consumed energy as the response time may be affected by the traffic distribution (which produces the differences in queuing time). The distribution of the random-based scheme shows smaller deviations as in this approach there is a high probability of choosing a server with a small (or none) queue occupancy. This scheme keeps the response time small but at the expense of provisioning a large number of active servers (and large energy consumption). More importantly, the task response time of the proposed task-scheduling scheme is bounded.

Figure 6 shows the histogram of the average task response time (total task response time divided by the number of tasks) for the proposed and random-based task-scheduling schemes. We observe a similar increase of average task response time for the proposed task-scheduling scheme to that of the random-based task-scheduling scheme.

We evaluated the energy consumption of the proposed and the random-based task scheduling schemes with respect to the number of servers under homogenous ($N = 1$) and exponentially-distributed task arrivals with a mean of 1000 tasks. Here, server capacity is bounded, $B_j = 40$. Some preliminary results of the proposed scheme were presented in [27]. In addition, we evaluated the average response time with respect to the number of servers in the data center of both schemes (Figure 7). The average task response time decreases as the number of servers increases in the random-based scheme as this scheme requires a larger number of servers in active

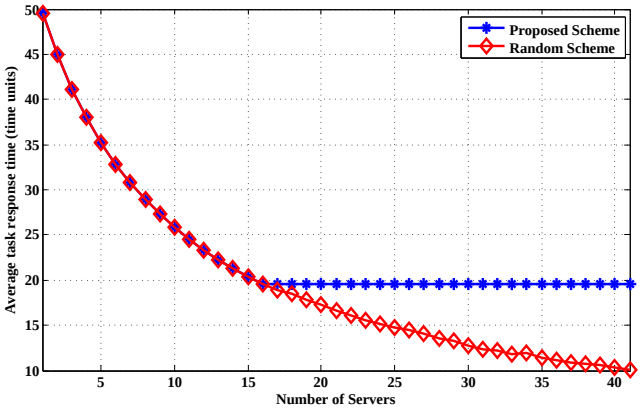

Fig. 7. Average task response time vs. number of servers, $B_j = 40$.

state and distributes tasks across all servers with uniform probability, unaware of energy savings. On the other hand, the average task response time decreases as the number of servers increases in the proposed scheme. Once the number of servers reaches an optimum, they handle tasks within the constrained task response time, such that this response time no longer increases. In this case, increasing the number of servers beyond the optimum number has no additional benefit. Therefore, turning off unnecessary servers can reduce additional energy consumption. On the other hand, the larger number of servers added in the random-based scheme results in larger energy consumption and resource over-provisioning.

V. Conclusions

In this paper, we propose a greedy task-scheduling scheme for a cloud-computing data center to reduce energy consumption by minimizing the number of servers while maintaining a deadline-based service-level agreement. We use integer programming optimization to minimize energy consumption and task response time in a data center handling heterogeneous tasks. We demonstrate that the number of servers needed by a data center to comply with those objectives is bounded through the use of the proposed greedy scheduling scheme. The proposed scheme minimizes the average task response time and, at the same time, minimizes server-related energy expenditure. We evaluate the proposed scheme using Matlab simulation with independent exponential distributed task arrivals and compare the performance of the proposed scheme with that of a random-based task-scheduling scheme. Simulation results show that a data center using the proposed task-scheduling scheme consumes on average over 70 times less on server energy than a data center using a random-based task-scheduling scheme. The proposed scheme saves energy at the cost of longer task response times, albeit within the task deadlines.

References

[1] "Google docs," http://docs.google.com.

[2] "Amazon web services," http://aws.amazon.com.

[3] "Azure services platform," http://www.microsoft.com/azure.

[4] "IBM smart business services," http://www.ibm.com/ibm/cloud.

[5] C. D. Patel and A. J. Shah, "Cost model for planning, development and operation of a data center," http://www.hpl.hp.com/techreports/2005/HPL-2005-107R1.pdf.

[6] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," in *Proceedings of the IEEE*, vol. 99, no. 1, January 2011, pp. 149 –167.

[7] "Power, pollution and the internet." [Online]. Available: http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html

[8] "Data center energy consumption trends." [Online]. Available: http://www1.eere.energy.gov/femp/program/dc_energy_consumption.html

[9] C. Clos, "A study of non-blocking switching networks," in *Bell Systems Technical Journal*, 1953, pp. 406–424.

[10] E. Oki, Z. Jing, R. Rojas-Cessa, and H. Chao, "Concurrent round-robin-based dispatching schemes for Clos-network switches," in *IEEE/ACM Trans. on Networking.*, vol. 10, no. 6, 2002, pp. 830–844.

[11] K. Pun and M. Hamdi, "Dispatching schemes for clos-network switches," *Computer Networks*, vol. 44, no. 5, pp. 667–679, 2004.

[12] C. Minkenberg and M. Gusat, "Speculative flow control for high-radix datacenter interconnect routers," in *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*. IEEE, 2007, pp. 1–10.

[13] M. Hamdi, "Building next generation massive data centers," *Security-Enriched Urban Computing and Smart Grid*, pp. 11–11, 2011.

[14] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: saving energy in data center networks," in *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, 2010, pp. 17–17.

[15] C. Minkenberg, R. Luijten, F. Abel, W. Denzel, and M. Gusat, "Current issues in packet switch design," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 119–124, 2003.

[16] "Data center energy consumption trends." [Online]. Available: http://www1.eere.energy.gov/femp/ program/dc_energy_consumption.html

[17] A. Bohra and V. Chaudhary, "Vmeter: Power modelling for virtualized clouds," in *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, April 2010, pp. 1 –8.

[18] I. Goiri, F. Juli and, R. Nou, J. Berral, J. Guitart, and J. Torres, "Energy-aware scheduling in virtualized datacenters," in *Cluster Computing (CLUSTER), 2010 IEEE International Conference on*, sept. 2010, pp. 58 –67.

[19] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 826–831. [Online]. Available: http://dx.doi.org/10.1109/CCGRID.2010.46

[20] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "Energy aware network operations," in *INFOCOM Workshops 2009, IEEE*, April 2009, pp. 1 –6.

[21] Q. Tang, S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," in *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 11, Nov. 2008, pp. 1458 –1472.

[22] B. Shi and A. Srivastava, "Thermal and power-aware task scheduling for hadoop based storage centric datacenters," in *Green Computing Conference, 2010 International*, Aug. 2010, pp. 73 –83.

[23] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No "power" struggles: coordinated multi-level power management for the data center," in *Proceedings of the 13th international conference on Architectural support for programming languages and operating systems*, ser. ASPLOS XIII. New York, NY, USA: ACM, 2008, pp. 48–59. [Online]. Available: http://doi.acm.org/10.1145/1346281.1346289

[24] A.-H. Mohsenian-Rad and A. Leon-Garcia, "Optimal residential load control with price prediction in real-time electricity pricing environments," *Smart Grid, IEEE Transactions on*, vol. 1, no. 2, pp. 120 –133, sept. 2010.

[25] A. Tam, K. Xi, and H. Chao, "Use of devolved controllers in data center networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*. IEEE, 2011, pp. 596–601.

[26] S. Khan and I. Ahmad, "Non-cooperative, semi-cooperative, and cooperative games-based grid resource allocation," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, april 2006, p. 10 pp.

[27] N. Liu, Z. Dong, and R. Rojas-Cessa, "Task and server assignment for reduction of energy consumption in datacenters," in *the 11th IEEE International Symposium on Network Computing and Applications, IEEE NCA12*, August 2012, pp. 1–4.