2012 IEEE 11th International Symposium on Network Computing and Applications

Task and Server Assignment for Reduction of Energy Consumption in Datacenters

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

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

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

***Abstract*—Energy consumption of cloud datacenters accounts for a major operational cost. This paper presents an optimization model for task scheduling to minimize task processing time and energy consumption in datacenters for cloud computing. We formulate an integer programming optimization problem to minimize the expected energy consumption of homogenous tasks in a datacenter with a large number of servers and propose the most-efficient-server first greedy task scheduling algorithm to minimize energy expenditure. We show that the proposed task scheduling can minimize the energy expenditure while bounding the average task waiting time. We present a simulation of the proposed task scheduling scheme to show an optimum number of servers to achieve small task processing times and to minimize energy consumption.**

***Keywords*-Cloud computing; Energy; Green Cloud; Task Scheduling;**

I. Introduction

Cloud computing has risen as a new computing paradigm that brings great flexibility and access to shared and scalable computing [1]–[4]. Cloud services are usually implemented in one or more datacenters where a large number of servers, storage units, and telecommunications infrastructure are provisioned. Energy expenditure of such datacenters contributes to a major portion of the operating cost [5], [6].

A lot of research has been done to study energy-aware cloud management and resource allocation schemes to reduce the energy consumption of datacenters [7]–[17]. One approach to reduce energy consumption is to reduce the number of startup servers. Incorporation of network traffic management and server workload consolidation has been investigated [10]. Schemes targeting an indirect detection of power dissipation, such as temperature distribution in the datacenter have been explored [11]. These schemes work as closed-loop systems where the sensed temperature is used to determine cool zones and to where tasks can be assigned.

An approach focusing on generating utility function using pricing models for real-time electricity was recently proposed [18]. This method exploits the pricing difference of electricity at different locations and times to control the load of the datacenters to optimize profit. Contributing factors such as server and virtual machine configuration in a cloud computing environment has been studied, however; without considering the energy efficiency of different virtual-machine configurations in the cloud [19].

Several scheduling schemes with focus on reducing energy consumption in cloud datacenters have been proposed [12]–[15]. A study on allocation of virtual servers to reduce datacenter energy consumption while keeping quality of service was performed [12]. Provisioning and configuration of virtual machines in datacenters was studied to observe the impact of different virtual machine configurations on datacenter energy consumption [13], [14]. A thermal and power-aware task scheduling for hadoop-based storage centric datacenters was proposed to ensure the nodes in the datacenter operate at a temperature below a certain threshold to reduce the power needed by the cooling system [15]. Dynamic resource allocation and power management in virtual datacenters using Lyapunov optimization was proposed to investigate the impact of resource allocation and power management with time-varying workloads and heterogeneous applications [16].

In this paper, we model the datacenter energy consumption in terms of task processing time and the number of servers required to meet expected task processing deadlines as an integer programming optimization problem. We show that for an optimized consumed energy in constrained-time task there is a bound on the maximum number of allocated servers. In addition, we propose a greedy task scheduling algorithm to minimize the energy consumption while keeping a bounded average task waiting time, which is defined by the service level agreement. The proposed greedy scheme, called the most-efficient-server first algorithm, assigns tasks to the most energy-efficient servers. Simulation results show that for a given task datacenter load, an optimum number of servers can be provisioned to minimize the average tasking waiting time and energy consumption.

The remainder of this paper is organized as follows. Section II presents the model of the datacenter adopted in this paper. Section III presents the proposed model of datacenter energy consumption and the most-efficient-server first task scheduling. Section IV presents simulation results of the proposed energy models and task scheduling schemes. Section V presents our conclusions.

II. Datacenter Model

A datacenter houses a large number (e.g., hundreds to thousands) of servers and storage units, which are interconnected through a network with diverse numbers of switches/routers and links, in an arrangement that may resemble a fat-tree topology. We assume that the network infrastructure provides enough bandwidth to avoid delays in the transmission of data and that most of the queuing processes occur at the servers.

In the cloud-computing datacenter, each server may be assigned to perform different or similar functions. Virtualization technologies allow the creation of multiple virtual

978-0-7695-4773-2/12 $26.00 © 2012 IEEE
DOI 10.1109/NCA.2012.42

Table I
TERMINOLOGY DEFINITION

Terminology	Definition
N	Number of task types
n_i	Number of type-i tasks
M	Number of servers
S_j	Server j, $1 \leq j \leq M$
T_w	Average waiting time per task
$x_{i,j}$	Number of task i assigned to S_j
$t_{i,j}$	Average processing time of type-i task on S_j
E	Total amount of energy in the datacenter
$P_{i,j}$	Power consumption of S_j to complete a type-i task
$B_{i,j}$	Capacity to store type-i tasks at S_j
$w_{i,j}$	Number of waiting type-i tasks on S_j
$\tau_{i,j}$	Average queueing delay of type-i tasks on S_j
X_j	Processing task schedule on S_j
ω	Weight vector

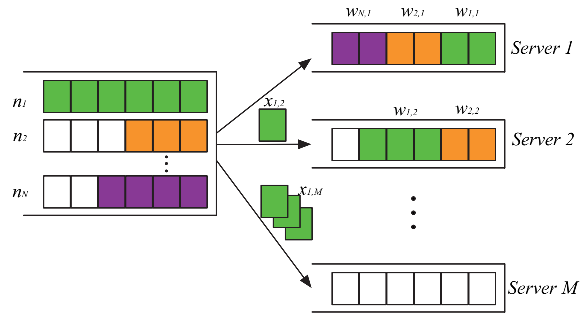

Figure 1. Model example of the datacenter task scheduler.

hosts on any of the available servers. Therefore, a task can be flexibly assigned to any server.

Servers can be modeled as a system that consumes energy in idle state to perform maintenance functions and to have all the subsystems ready while it waits for task to arrive [20]. Once a task arrives, a server processes the task and it may spend an additional amount of energy, which depends on the number of resources demanded by the task.

In the following section, we model the cloud datacenter task scheduling as an integer linear programming optimization problem and introduce the most-efficient-server first algorithm to minimize average task waiting time and energy consumption.

III. TASK SCHEDULING AND ENERGY CONSUMPTION

We formulate the energy consumption of the datacenter as an optimization problem for minimizing the total energy expenditure based on minimizing the number of servers needed. The definitions of terminology used in the remainder of this paper are listed in Table I.

A. Optimization of processing time and number of busy servers

A datacenter is required to handle tasks that require different computational resources. Therefore, servers may provide different processing times (assumed that a complete task is assigned to a single server) and different levels of energy consumption. In this paper, we assume there are N different types of serviceable tasks in the datacenter. The number of tasks of type i is n_i, where $1 \leq i \leq N$. The number of servers in a datacenter is M, and each server is denoted as S_j, where $1 \leq j \leq M$..

Here, $B_{i,j}$ denotes the number of type-i tasks S_j can queue and process. Therefore, this is also the maximum number of type-i tasks that a datacenter scheduler can allocate to S_j, $x_{i,j}$ denotes the number of type-i tasks assigned to S_j, T_w denotes the average waiting time per task, $t_{i,j}$ denotes the time S_j takes to process a type-i task, and $w_{i,j}$ denotes the number of type-i tasks queued at S_j. Figure 1 shows an example of the task scheduler, where the queues represent the servers.

Herein, $P_{i,j}$ denotes the power consumed by S_j to complete a type-i task and $\tau_{i,j}$ denotes the time a type-i task has to be queued to be serviced by S_j, X_j denotes the task schedule vector for S_j, and ω as the weight vector. Here, X_j is defined as an $N \times \sum_{i=1}^{N} x_{i,j}$ matrix with elements 0 and 1 to represent the scheduled sequence of each task. The rows the matrix indicate the task type and the columns indicate the sequence. 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 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 in the column.

The total amount of energy the datacenter consumes, denoted as E, is defined as the sum of energy consumed by the processing of all the tasks at the datacenter in a defined period of time. Consider the power consumption for each type of task, the number of task, and the time to process the task, the total consumed energy is calculated as

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

The objective function is to find an optimum assignment $x_{i,j}$ such that the time the 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 datacenter has not received any task initially, therefore the upcoming tasks are immediately assigned to the servers, and 2) when the datacenter has backlogged tasks in it, such that upcoming tasks have to be queued and wait for service (assignment).

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

$$\begin{aligned} \min_{x} \quad & E(x) = \sum_{i=1}^{N} \sum_{j=1}^{M} t_{i,j} P_{i,j} x_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{M} x_{i,j} = n_i, x_{i,j} \leq B_{i,j} \end{aligned} \tag{2}$$

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

$(t_{1,j}, t_{2,j}, \cdots, t_{N,j})$, X_j is an $N \times \sum_{i=1}^{N} x_{i,j}$ matrix, and ω is a column vector $(\sum_{i=1}^{N} x_{i,j} - 1, \sum_{i=1}^{N} x_{i,j} - 2, \cdots, 1, 0)^T$.

2) With queueing delay: When the datacenter is fully loaded and upcoming tasks need to be queued to wait for service, the optimization problem is formulated as

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

The average waiting time is

$$T_w = (\sum_{j=1}^{M} \sum_{i=1}^{N} x_{i,j} \tau_{i,j} + T_j X_j \omega) / \sum_{i=1}^{N} n_i. \tag{4}$$

In the remainder of this paper, we focus on analyzing the assignment of a single task type ($N = 1$). This can be considered under the assumption that other types can be decomposed into the single type. To simplify the notation for this case, we remove the subscript i in the notation for $t_{i,j}$, $x_{i,j}$,$\tau_{i,j}$ in the remainder of this section. The average waiting time (4) is represented as $T_w = \sum_{j=1}^{M} \frac{t_j}{2} x_j (x_j - 1)$ where $\sum_{j=1}^{M} x_j = n$, where n is the number of tasks.

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

$$2n - (2B_j - 1) t_j \sum_{j=1}^{M} \frac{1}{t_j} \leq M \leq 2n - t_j \sum_{j=1}^{M} \frac{1}{t_j}. \tag{5}$$

We can also find the range, and therefore the bound, of n that can be allocated to the servers as defined by M, B_j and $\sum_{j=1}^{M} t_j$.

$$\frac{1}{2}(M + t_j \sum_{j=1}^{M} \frac{1}{t_j}) \leq n \leq \frac{1}{2}(M + (2B_j - 1) t_j \sum_{j=1}^{M} \frac{1}{t_j}) \tag{6}$$

B. The most-efficient-server first algorithm

Should servers with higher computing capacity be available, where the energy consumption is proportional to the server capacity, these servers become the most preferred ones. In this case, the optimization problem may be interpreted as a greedy-assignment algorithm. For this, it is considered that the scheduler sorts the servers based on energy efficiency and assigns task 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 become full. The greedy algorithm can be defined according to the following pseudo-code:

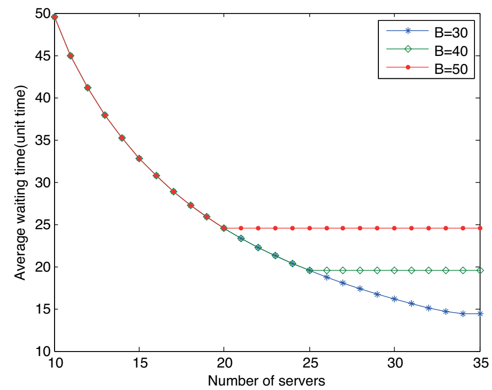

Figure 2. Average waiting time vs. number of servers.

Algorithm 1 The most-efficient-server first algorithm

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 S_j* **do**
    Calculate server energy consumption $E_{i,j} = P_{i,j} t_{i,j}$
    if *E(i,j) ≤ E(a,b)* **then**
      a = i b = j
    end
  end
end
Schedule(a to S_b)
while *unscheduled tasks remain* **do**
  for *Each S_j* **do**
    Calculate energy consumption E'(i,j) **if** *E'($y_{i'}$,j) ≤ E'($y_{i'}$,c)* **then**
      c = j
    end
  end
  Schedule(Task $y_{i'}$ to S_c)
end

IV. Simulation Results

We simulated the proposed greedy algorithm in Matlab under homogenous ($N = 1$) and exponentially-distributed task arrivals with a mean of n =1000 tasks. The evaluation is based on measuring the average task-waiting time and total energy consumed by the datacenter vs. the number of active servers to handle the tasks.

Figure 2 shows the average task-waiting time under different queuing capacity for $B = \{50, 40, 30\}$. As the number of servers assigned to handle the tasks increases, the average task waiting time decreases. However, there is a minimum achievable waiting time; this cannot longer be reduced once additional servers are added as there are no more tasks to be assigned or the active servers can continue providing service to upcoming tasks.. Therefore, the number of servers is bounded to $M = \{20, 25, 34\}$ for $B = \{50, 40, 30\}$, respectively.

To evaluate the average task waiting time under different input loads, the tasks arrivals with mean n =

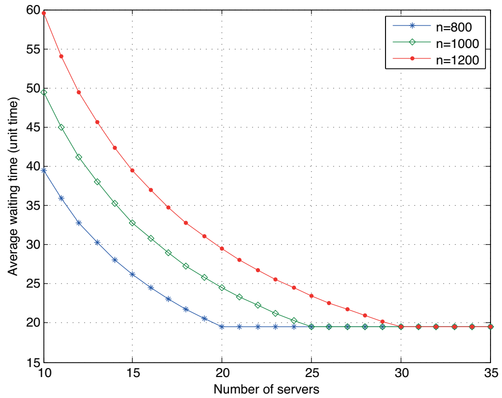

Figure 3. Average waiting time vs. number of servers.

$\{800, 1000, 1200\}$ and servers with queuing capacity of $B = 40$ were considered. Figure 3 shows the task average waiting time for different number of servers and under different loads (i.e., different number of tasks). As the number of tasks increases, the number servers required to keep the average task waiting time within the bound also increased. These results provide an insight to the provisioning of the number of servers based on a given number of tasks such that quality of service requirements, in terms of average waiting time, are met.

V. Conclusions

In this paper, we focus on task scheduling for a cloud datacenter using integer programming optimization to minimize energy consumption and task waiting time. We propose the most-efficient server first scheduling algorithm to distribute tasks among servers of a datacenter to minimize their average waiting time, and at the same time, minimize energy expenditure. We modeled the proposed algorithm in Matlab and simulated it under exponentially-distributed task arrivals. The results show that the number of servers required to achieve a minimum task waiting time is bounded. This information is useful to determine the largest number of servers that can be turned on for a given workload. We generalized the model for different task types and focused on single-type task.

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] 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.

[8] 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.

[9] 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

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

[11] 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.

[12] M. Taheri and K. Zamanifar, "2-phase optimization method for energy aware scheduling of virtual machines in cloud data centers," in *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for*, dec. 2011, pp. 525 –530.

[13] V. Raj and R. Shriram, "Power aware provisioning in cloud computing environment," in *Computer, Communication and Electrical Technology (ICCCET), 2011 International Conference on*, march 2011, pp. 6 –11.

[14] K. Bouyoucef, I. Limam-Bedhiaf, and O. Cherkaoui, "Optimal allocation approach of virtual servers in cloud computing," in *Next Generation Internet (NGI), 2010 6th EURO-NF Conference on*, june 2010, pp. 1 –6.

[15] 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.

[16] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, "Dynamic resource allocation and power management in virtualized data centers," in *PROC. IEEE/IFIP NOMS*, 2010.

[17] 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

[18] 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.

[19] G. Wen, J. Hong, C. Xu, P. Balaji, S. Feng, and P. Jiang, "Energy-aware hierarchical scheduling of applications in large scale data centers," in *Cloud and Service Computing (CSC), 2011 International Conference on*, dec. 2011, pp. 158 –165.

[20] R. Rojas-Cessa, S. Pessima, and T. Tian, "Experimental evaluation of energy savings of virtual machines in the implementation of cloud computing," in *Wireless and Optical Communications Conference (WOCC), 2012 21st Annual*, april 2012, pp. 65 –70.