Sustainable Computing: Informatics and Systems 6 (2015) 3–16

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Distributed temperature-aware resource management in virtualized data center

Mohammad A. Islam[a,*], Shaolei Ren[a], Niki Pissinou[a], A. Hasan Mahmud[a], Athanasios V. Vasilakos[b]

[a] Florida International University, Miami, FL 33199, United States
[b] National Technical University of Athens, Athens 106 80, Greece

ARTICLE INFO

Article history:
Received 16 November 2013
Accepted 14 March 2014

Keywords:
Capacity provisioning
Data center
Distributed optimization
Load distribution
Temperature constraint

ABSTRACT

While resource consolidation enables energy efficiency in virtualized data centers, it results in increased power density and causes excessive heat generation. To prevent servers from overheating and avoid potential damage and/or service outages, data centers need to incorporate temperature awareness in resource provisioning decisions. Moreover, data centers are subject to various peak power constraints (such as peak server power) that have to be satisfied at *all* times for reliability concerns. In this paper, we propose a novel resource management algorithm, called DREAM (Distributed REsource mAnagement with teMperature constraint), to optimally control the server capacity provisioning (via power adjustment), virtual machine (VM) CPU allocation and load distribution for minimizing the data center power consumption while satisfying the Quality of Service (QoS), IT peak power and maximum server temperature constraints. By using DREAM, each server can autonomously adjust its *discrete* processing speed (and hence, power consumption, too), and optimally decide the VM CPU allocation as well as amount of workloads to process in the hosted VMs, in order to minimize the total power consumption which incorporates both server power and cooling power. We formally prove that DREAM can yield the minimum power with an *arbitrarily* high probability while satisfying the peak power and server temperature constraints. To complement the analysis, we perform a simulation study and show that DREAM can significantly reduce the power consumption compared to the optimal temperature-unaware algorithm (by up to 33%) and equal load distribution (by up to 86%).

1. Introduction

The rapid expansion of Internet-based services, together with the demand for scalable and robust infrastructures, has resulted in thriving number and size of virtualized data centers in recent years. While virtualization offers improved "power proportionality" through resource consolidation, the increasing number of virtualized data centers still accounts for a huge electricity consumption, and raises serious economic and environmental concerns due to the accompanying high electricity bill and carbon footprint. Hence, data center operators are constantly urged to reduce the power consumption while maintaining a premium Quality of Service (QoS).

For a data center serving delay-sensitive workloads such as web services, lowering energy consumption without affecting the QoS is challenging. Although data center operation has attracted significant interest from the research community and undergone a substantial improvement [1], there still exist some hurdles that limit the data center operations. First, data centers have a stringent IT peak power budget: exceeding the peak power constraint will result in serious consequences such as service outages and equipment damages [2–4]. Increasing the power budget, however, may not be an immediate or viable solution in practice, as the peak power budget is often determined during the construction phase and capital cost of building a data center is directly proportional to the provisioned IT peak power (currently, estimated at 10–20 U.S. dollars per Watt) [4]. Thus, judiciously allocating the total power budget to different server units while considering the peak power constraint is crucial for optimizing the data center operation. Second, with the ever-increasing power density generating an excessive amount of heat, thermal management in data centers is becoming imperatively important for preventing server

* Corresponding author. Tel.: +1 3053482032.
E-mail addresses: misla012@fiu.edu (M.A. Islam), sren@fiu.edu (S. Ren), pissinou@fiu.edu (N. Pissinou), amahm008@fiu.edu (A.H. Mahmud), vasilako@ath.forthnet.gr (A.V. Vasilakos).

http://dx.doi.org/10.1016/j.suscom.2014.03.002
2210-5379/

overheating that could potentially induce server damages and huge economic losses [5]. As a consequence, optimally distributing the workloads to facilitate heat recirculation and avoid overheating has to be incorporated in data center operation. Last but not least, for system scalability, distributed resource management in data centers is highly desired, which, however, is not readily available in all scenarios.

In this paper, we propose a novel resource management algorithm, called DREAM (Distributed REsource mAnagement with teMperature constraint), to control the server capacity provisioning (via power adjustment), virtual machine (VM) CPU allocation and load distribution for minimizing the data center power consumption while satisfying the QoS, IT peak power and maximum server inlet temperature[1] constraints. Unlike temperature-*reactive* approaches that prevent server overheating based on the observed real-time temperature (e.g., shut down servers when they become *hot* [5,6]), DREAM makes distributed decisions while proactively taking into account the potential impact of the decisions on the inlet temperature increase to avoid server overheating. While total power is minimized for the data center operation by incorporating the server power and cooling system power in the optimization objective, the QoS constraint, quantified by average delay threshold, guarantees the overall service quality. By using DREAM, each server can autonomously adjust its *discrete* processing speed (and hence, power consumption, too) and optimally decide the VM CPU allocation and amount of workloads to process, in order to minimize the total power consumption. DREAM builds upon a variation of Gibbs sampling technique [7], combined with dual decomposition [8]. We conduct a rigorous performance analysis and formally prove that DREAM can yield the minimum power, while satisfying the QoS, peak power and server inlet temperature constraints. We also perform an extensive simulation study to complement the analysis. The simulation results are consistent with our theoretical analysis. Moreover, we compare DREAM with two existing algorithms and show that DREAM reduces the power by up to 33%.

Our main contributions are summarized as follows:

1. We develop a distributed algorithm, DREAM, for data centers in which each server can autonomously decide its processing speed, CPU allocation for the hosted VMs and the amount of workloads to process, while maintaining QoS and incorporating three important practical constraints: discrete processing speeds, peak server power, and maximum server inlet temperature constraints. It is rigorously proved that DREAM minimizes the total power with an arbitrarily high probability.
2. We conduct a comprehensive simulation, and the results show that DREAM achieves a significantly lower power consumption compared to two widely used existing algorithms.

The rest of this paper is organized as follows. The model is described in Section 2. In Section 3 and 4, we present the problem formulation and develop our distributed online algorithm, DREAM, respectively. Section 5 provides a simulation study to validate DREAM. Related work is reviewed in Section 6, and finally, concluding remarks are offered in Section 7.

2. Model

We consider a model in which the capacity provisioning, VM CPU allocation and workload distribution decisions are updated periodically and the period is sufficiently large (e.g., 1 h) such that the room temperature can become stabilized. Moreover, at the beginning of each decision period, the data center operator can predict the workload arrival rate over the period. For example, each decision period corresponds to 1 h if the data center leverages hour-ahead workload arrival prediction that is readily available in practice [4,9,10]. Throughout the paper, we drop the time index wherever applicable without affecting the analysis. Next, we present the modeling details for the data center and workloads. Key notations are summarized in Table 1.

Table 1
List of notations.

Notation	Description
x_i	Speed of server i
$c_{i,j}$	CPU allocation for $\text{VM}_{i,j}$
$\mu_{i,j}$	Service rate of $\text{VM}_{i,j}$
$\lambda_{i,j}$	Workloads distributed to $\text{VM}_{i,j}$
p_i	Average power consumption of server i
p	Total power consumption
$d_{i,j}$	Average delay in $\text{VM}_{i,j}$
T_{sup}	Supply temperature
T_{out}	Outside air temperature
$\mathbf{T}_{\text{in}}$	Server inlet temperature
$\mathbf{D}$	Heat transfer matrix
N	Number of servers
M	Number of server racks
J	Number of job types

2.1. Data center

We consider a data center that has N physical servers that are mounted in M racks (or chassis). Each server hosts multiple VMs to serve different types of workloads. Without causing ambiguity, we also use *servers* to represent physical servers wherever applicable. The mth rack contains n_m servers such that $\sum_{m=1}^{M} n_m = N$. We denote the entire set of servers and the subset of servers mounted in the mth rack by $\mathcal{N} = \{1, 2, \ldots, N\}$ and $\mathcal{N}_m$, respectively, and it follows naturally that $\mathcal{N} = \mathcal{N}_1 \bigcup \mathcal{N}_2 \bigcup \ldots \bigcup \mathcal{N}_M$ and $\mathcal{N}_i \bigcap \mathcal{N}_j = \varnothing$ if $i \neq =j$. In general, the servers are heterogeneous in their power consumptions and processing speeds due to various reasons such as different purchase dates. Moreover, each server may trade performance for power consumption by varying its performance and power states (e.g., P-states, C-states, or a combination of them), varying its processing speed (e.g., via dynamic voltage and frequency scaling or DVFS [11]), or switching servers on/off. In our study, we only focus on the cooling power and server power for the considered workloads, while neglecting the power consumption of other parts (e.g., power supply system) which, however, can be conveniently absorbed by a (partial) power usage effectiveness (PUE) factor [9].

2.1.1. Server power

As computing takes up a large portion (typically 40%) of server power consumption [3], adjusting CPU speed can significantly affect the total power consumption. Hence, we focus on CPU resource allocation, while treating other resources (e.g., memory, disk) as sufficient and *non-bottleneck* resources. Although this assumption may not hold for all application scenarios (e.g., memory/disk power consumption may vary considerably for I/O-intensive workloads), we note that it is reasonably accurate for CPU-intensive workloads that are the main concentration of our study [12,13].

To keep our model general, we consider that server i can choose its speed x_i out of a finite set $S_i = \{s_{i,0}, s_{i,1}, \ldots, s_{i,L_i}\}$, where $s_{i,0} = 0$ represents zero speed (server deep sleep or shut down) and L_i is the number of available positive speed settings. The speed x_i quantifies

[1] Server inlet temperature is the temperature of air entering/exiting the server, and it is different from the server component temperature which is handled using a separate mechanism by the server itself (e.g., fan speed increases if the CPU temperature increases).