2013 IEEE Sixth International Conference on Cloud Computing

Minimizing the Operational Cost of Data Centers via Geographical Electricity Price Diversity

Zichuan Xu Weifa Liang
Research School of Computer Science, The Australian National University
Canberra, ACT 0200, Australia
Email: edward.xu@anu.edu.au, wliang@cs.anu.edu.au

***Abstract*—Data centers, serving as infrastructures for cloud services, are growing in both number and scale. However, they usually consume enormous amounts of electric power, which lead to high operational costs of cloud service providers. Reducing the operational cost of data centers thus has been recognized as a main challenge in cloud computing. In this paper we study the minimum operational cost problem of fair request rate allocations in a distributed cloud environment by incorporating the diversity of time-varying electricity prices in different regions, with an objective to fairly allocate requests to different data centers for processing while keeping the negotiated Service Level Agreements (SLAs) between request users and the cloud service provider to be met, where the data centers and web portals of a cloud service provider are geographically located in different regions. To this end, we first propose an optimization framework for the problem. We then devise a fast approximation algorithm with a provable approximation ratio by exploiting combinatorial properties of the problem. We finally evaluate the performance of the proposed algorithm through experimental simulation on real-life electricity price data sets. Experimental results demonstrate that the proposed algorithm is very promising, which not only outperforms other existing heuristics but also is highly scalable.**

I. Introduction

With the rapid development of processing and storage technologies and the success of the Internet, computing resources have become cheaper, more powerful and more ubiquitously available than ever before. This technological trend has enabled the realization of a new computing model called cloud computing, in which resources (e.g., software, platforms and infrastructures) are provided as general utilities that can be leased and released by users through the Internet in an on-demand fashion. The emergence of cloud computing has made a tremendous impact on the Information Technology (IT) industry over the past few years, where large-scale data centers have been built in different regions by cloud service providers [8], [10]. Although cloud computing benefits users by freeing them from setting and maintaining IT infrastructures, it increases the operational cost of cloud service providers due to the large quantity of electricity consumption. According to a McKinsey report [11], a typical data center consumes as much energy as 25,000 households per year, and the electricity bill for data centers in 2010 is estimated over $11 billion and this cost is almost doubled every five years, and the electricity cost by major cloud service providers is 30%-50% percentage of their total operational cost [15]. To minimize the operational cost of data centers, lots of efforts have been taken recently, and different approaches have been developed which include Dynamic Voltage and Frequency Scaling (DVFS), dynamic sizing data centers, and exploiting electricity price diversities. Following the electricity market policies, the electricity prices vary over time (e.g., they change from every 15 minutes to every one hour), and usually are determined by the clearing prices of the current supplies and demands. This creates an opportunity for cloud service providers to reduce their operational costs through dynamically allocating user requests to these data centers with much cheaper electricity at that moment. However, shifting user requests and the results of the requests between the web portals and the data centers consume a large amount of network bandwidth, and the cost of this bandwidth consumption must be taken into account [21]. In addition to the costs of electricity and network bandwidth consumption, it is very important for a cloud service provider to provide quality services in order to avoid the penalties of unfulfilled SLA agreements, thereby reducing the revenue loss of the cloud service provider. In this paper we will use the average service delay of each request as its QoS requirement, and refer to it as the *delay requirement*. Specifically, we deal with a fundamental operational cost optimization problem in a distributed cloud computing environment with an aim of maximizing the system throughput while minimizing the operational cost and satisfying user SLAs, through exploring electricity price diversities.

The main contributions of this paper are summarized as follows. We consider minimizing the operational cost of a distributed cloud service provider whose data centers are geographically located in different regions. We first formulate a novel optimization problem, namely the minimum operational cost problem for fair request rate allocations. We then develop a combinatorial approximation algorithm with a provable approximation ratio, by reducing the problem to a minimum cost multicommodity flow problem. We finally conduct extensive experiments by simulation to evaluate the performance of the proposed algorithm, using real-life electricity price data sets. Experimental results demonstrate that the proposed algorithm is effective and promising.

To the best of our knowledge, this is the first fast approximate solution to the minimum operational cost problem in distributed cloud environments by exploiting combinatorial properties of the problem. The proposed algorithm is not only highly scalable but also running fast in response to dynamic

978-0-7695-5028-2/13 $26.00 © 2013 IEEE
DOI 10.1109/CLOUD.2013.94

changes of electricity prices and request rates. In contrast, most existing solutions formulated the problem or the similar optimization problems as a mixed integer programming (MIP) and solved the MIP through the randomness rounded technique or the other heuristics. Thus, the accuracy of their solution is not guaranteed, or is achieved at the expense of expensive computation time. For the latter, even such an exact solution is found, it may not be applicable due to the time-varying nature of electricity prices and user request rates.

The remainder of the paper is organized as follows. Section II briefly introduces related works, followed by introducing the system model, the problem definition, and a well-known approximation algorithm in Section III. An optimization framework and an approximation algorithm for the problem are proposed in Section IV. Section V conducts experiments to evaluate the performance of the proposed algorithm through simulations, and the conclusion is given in Section VI.

II. Related work

A global cloud service provider (e.g. Google) usually deploys many data centers in different geographical regions for resource saving and convenience. To achieve energy efficiency for such cloud service providers, a promising approach is to explore the time-varying electricity prices in these regions where the data centers are located at. However, minimizing the energy consumption of geographically distributed data centers is essentially different from that of a single data center, this poses a new challenge to design efficient algorithms for energy management in such geographically distributed data centers, and several efforts have been taken in the past several years [15], [2], [16], [17], [6], [12], [20], [13], [14]. For example, Qureshi *et al.* [15] initialized the study by characterizing the energy expense per unit of computation due to fluctuating electricity prices. They empirically showed that the exploration of electricity price diversity may save millions of dollars per day. Buchbinder *et al.* [2] considered the energy optimization through migrating jobs among data centers and devised an online algorithm to reduce the electricity bill. Guo *et al.* [6] made use of the Lyapunov optimization technique to reduce the electricity bill of data centers, using temporary energy storages like UPS. Le *et al.* [12] devised a general framework to manage the usage of "brown energy" (produced via carbon-intensive means) and "green" energy (renewable energy) with the aim of reducing environmental effects on huge amount of energy consumed by the data centers. Zhang *et al.* [20] investigated the problem of geographical request allocation to maximize the usage of renewable energy under a given operation budget. Liu *et al.* [13], [14] dealt with the energy sustainability of data centers by exploring sources of renewable energy and geographical load-balancing. They demonstrated the necessity of the storage of renewable energies. The most closely related studies to the work in this paper are due to Rao *et. al* [17], [16], [18]. They [17], [16] considered the minimum operational cost problem in distributed data centers through exploring the diversity of electricity prices, by formulating the problem as a mixed integer programming (MIP) and providing a heuristic to the MIP. They also proposed a flow-based algorithm for the MIP [18].

Most existing studies in literature on geographical request allocation subject to diverse electricity prices and bandwidth costs focused on solving a constrained optimization problem with one or multiple constraints [17], [16], [18], [20]. However, finding an exact solution usually takes a much longer time due to highly computational complexity. The solution based on the MIP thus is only suitable for a small or medium network size and not scalable. Even if such a solution is found, it may not be applicable in the reality due to time varying nature of both electricity prices and request rates in the system.

III. Preliminaries

In this section we first introduce the system model and notations. We then define the problem precisely. We finally introduce a fast approximation algorithm for the minimum cost multicommodity flow problem which will be used later.

A. System model

We consider a distributed cloud computing environment that consists of a set of geographically distributed data centers $\mathcal{DC} = \{DC_i \mid 1 \leq i \leq N\}$ and a set of web portals $\mathcal{WP} = \{WP_j \mid 1 \leq j \leq M\}$. Each data center DC_i is equipped with hundreds of thousands of homogeneous servers with $N_i = |DC_i|$, and denote by μ_i the service rate of each server in DC_i for each i, where $1 \leq i \leq N$. Each web portal WP_j serves as a front-end server that directly receives requests from the users, performs request rate allocations to data centers. Each data center and every web portal can communicate with each other through the Internet. Each user sends its requests to its nearby web portal and the requests are then forwarded to different data centers. The requests allocated to a data center are initially stored in its $M/M/n$ waiting queue [9] prior to being processed. To respond to time-varying request rates and electricity prices, the time is assumed to be slotted into equal *time slots*. The request rate allocation will be performed at each time slot.

Let $r_j \in \mathbb{Z}$ be the request rate at web portal WP_j at a time slot t. For the sake of convenience, in this paper we assume that each time slot is one hour so that the servers in data centers will not be turned on and off quite often, given the significant wear-and-tear cost of power-cycling. Associated with each user request, there is a tolerant delay requirement which in certain extent represents the negotiated SLA between the user and the cloud service provider.

Given a data center DC_i containing N_i servers with the average service rate μ_i per server, denote by $r_{j,i} \in \mathbb{Z}$ the request rate from web portal WP_j to data center DC_i. Let Pr_i be the probability of requests waiting in the queue, then the average delay of a request in DC_i is [17], [13], [14]

$$D_i(N_i, \sum_{j=1}^{M} r_{j,i}) = \frac{Pr_i}{N_i\mu_i - \sum_{j=1}^{M} r_{j,i}}, \tag{1}$$

where we assume that there always are requests in the waiting queue of DC_i, i.e., $Pr_i = 1$. To meet the negotiated SLA of each request in DC_i, the average waiting time of all requests in it is no greater than the user tolerant delay. Otherwise, it will incur the late penalty due to the violation of the specified SLA. Worst of all, the users may no longer use the cloud service provided by the cloud service provider in the future.

B. Electricity cost model

The electricity markets typically have the structures of both regulated utility and deregulated wholesale. In the regulated electricity market, the electricity price is fixed during a day and may have on-peak and off-peak prices, while in the deregulated electricity market, the electricity price varies over time. Denote by $p_i(t)$ the electricity price per unit energy at time slot t at which data center DC_i is located. The electricity cost incurred by data center DC_i for the duration of time slot t is determined by the amount of power it consumed and the electricity price. Let $E_i(t)$ be the total energy consumption of data center DC_i at time slot t. Assuming that the load among the servers in each data center is well balanced, $E_i(t)$ thus is proportional to the energy consumption per request in DC_i. Let α_i be the average energy consumed per request in it which is a constant. Then the power consumption in data center DC_i at time slot t is

$$E_i(t) = \alpha_i \sum_{j=1}^{M} r_{j,i}, \tag{2}$$

and the electricity cost at data center DC_i at time slot t is

$$p_i(t)E_i(t) = p_i(t)\alpha_i \sum_{j=1}^{M} r_{j,i}. \tag{3}$$

C. Network bandwidth cost

Most existing studies assumed that allocating requests to data centers is transparent and does not incur any cost on the communication bandwidth usage, as stateless requests from the web portals normally consist of simple jobs and can be enclosed by small data packages. However, a cloud service provider may lease the bandwidth from an ISP for its data transfer between the web portals and the data centers, a charge of occupying the bandwidth during the acquisition or construction phase of communication links (e.g. TCP/IP links) will be applied [21]. We thus assume that the bandwidth consumption cost is proportional to the number of requests. Since the cloud service provider may lease bandwidths with different data transfer rates between web portals and data centers according to the request loads at web portals, the cost of bandwidth consumption by transferring a request from different web portals is different. Let p_j^b denote the bandwidth cost of transferring a single request from WP_j to any data center. Then, the total cost of transferring requests from web portal server WP_j to all data centers is

$$p_j^b \sum_{i=1}^{N} r_{j,i}. \tag{4}$$

D. Problem definition

The minimum operational cost problem for fair request rate allocations for the cloud service provider at time slot t is to allocate a fractional request rate $r_{j,i} \in \mathbb{Z}$ from each web portal WP_j to each data center DC_i such that the operational cost $\sum_{i=1}^{N} p_i(t)\alpha_i \sum_{j=1}^{M} r_{j,i} + \sum_{j=1}^{M} p_j^b r_{j,i}$ is minimized while the system throughput $\sum_{i=1}^{N}\sum_{j=1}^{M} r_{j,i}$ is maximized (or equivalently the number of servers n_i at each DC_i to be switched on) and all allocated requests are served within their SLAs, subject to that $\sum_{i=1}^{N} r_{j,i} \le r_j$ and $0 \le n_i \le N_i$ for all i and j with $1 \le i \le N$ and $1 \le j \le M$. In other words, the objective is to make each web portal have the same proportion of number of requests to be served and this proportion λ is as large as possible while the associated cost is minimized, i.e., to maximize $\sum_{j=1}^{M} \lambda \cdot r_j = \sum_{i=1}^{N}\sum_{j=1}^{M} r_{j,i}$ where $0 \le \lambda \le 1$.

E. The minimum cost multicommodity flow problem

Given a directed graph $G(V, E; u, c)$ with capacities $u : E \mapsto R^+$ and costs $c : E \mapsto R^{\ge 0}$, assume that there are k source-destination pairs $(s_i, t_i; d_i)$ where d_i is the amount of demands to be routed from its source node s_i to its destination node t_i for all i with $1 \le i \le k$. Let B (> 0) be a given budget and $|f_i|$ the amount of flow f_i sent from s_i to t_i. The *minimum cost multicommodity flow problem* in G is to find a largest λ such that there is a multicommodity flow f_i routing $\lambda \cdot d_i$ units of commodity i from s_i to t_i for each i with $1 \le i \le k$, subject to the flow constraint and the budget constraint $\sum_{e \in E} c(e) \cdot f(e) \le B$, where $f(e) = \sum_{i=1}^{k} f_i(e)$.

The optimization framework given by Garg and Könemann is as follows. It first formulates the problem as a linear programming **LP**, then finds an approximate solution to the duality **DP** of **LP** that returns an approximate solution to the original problem. Let $\mathcal{P}_i$ be the set of paths in G from s_i to t_i, and let $\mathcal{P} = \cup_{i=1}^{k} \mathcal{P}_i$. Variable f_p represents the flow on path $p \in \mathcal{P}$. The linear programming formulation of the problem **LP** is

$$\begin{array}{lll} \textbf{LP:} & \max & \lambda \\ & \textit{s.t.} & \sum_{e \in p} f_p \le u(e) \qquad\qquad \forall e \in E, \\ & & \sum_{i=1}^{L} \sum_{p \in \mathcal{P}_i} \sum_{e \in p} (f_p \cdot c(e)) \le B, \\ & & \sum_{p \in \mathcal{P}_i} f_p \ge \lambda \cdot d_i \qquad \forall i, 1 \le i \le k, \\ & & f_p \ge 0 \qquad\qquad \forall p \in \mathcal{P}, \\ & & 0 \le \lambda \le 1. \end{array}$$

The dual linear programming **DP** of **LP** is described as follows, where $l(e)$ is the *length* on every edge $e \in E$ and ϕ is viewed as the *length* of the cost constraint B.

$$\begin{array}{lll} \textbf{DP:} & \min & D(l, \phi) = \sum_{e \in E} l(e)u(e) + B \cdot \phi, \\ & \textit{s.t.} & \sum_{e \in p} (l(e) + c(e) \cdot \phi) \ge z_i \qquad \forall p \in \mathcal{P}_i, \\ & & \sum_{i=1}^{k} d_i \cdot z_i \ge 1, \\ & & l(e) \ge 0 \qquad\qquad \forall e \in E, \\ & & z_i \ge 0 \qquad\qquad \forall i, 1 \le i \le k. \end{array}$$

Specifically, Garg and Könemann's optimization framework for the **DP** proceeds in a number of *phases*, while each phase is composed of exactly k iterations, corresponding to the k

commodities. Within each iteration, there are a number of *steps*. Initially, $l(e) = \frac{\delta}{u(e)}$ for each $e \in E$, $\phi = \delta/B$, $z_i = \min_{p \in \mathcal{P}_i}\{l(p) + c(p)\phi\}$, where $c(p) = \sum_{e \in p} c(e)$, $\delta = \left(\frac{1-\epsilon}{|E|}\right)^{1/\epsilon}$, and ϵ is the increasing step of the length function in each iteration. In one phase, let i be the current iteration, the algorithm will route d_i units of flow of commodity i from s_i to t_i within a number of steps. In each step, it routes as much fraction of commodity i as possible along a shortest path p from s_i to t_i with the minimum $l(p) + c(p)\phi$. The amount of flow sent on p is the minimum one u_{min} among the three values: the bottleneck capacity of p, the remaining demand d_i' of commodity i and $B/c(p)$. Once the amount of the flow u_{min} has been routed on p, the dual variables l and ϕ are then updated: $l(e) = l(e)(1 + \epsilon\frac{u_{min}}{u(e)})$ and $\phi = \phi(1 + \epsilon\frac{u_{min} \cdot c(p)}{B})$. The algorithm terminates when the value of the objective function $D(l, \phi) \geq 1$. A feasible flow is finally obtained by scaling the flow by $\log_{1+\epsilon}\frac{1}{\delta}$ [7].

Theorem 1: (see [7]) There is an approximation algorithm for the minimum cost multicommodity flow problem in $G(V, E; u, c)$ with k commodities to be routed from their sources to their destinations, which delivers a solution with an approximation ratio of $(1-3\epsilon)$ while the associated cost is the minimum. The algorithm takes $O^*(\epsilon^{-2}m(k+m))$ time[1], where $m = |E|$ and ϵ is a given constant with $0 < \epsilon \leq 1/3$.

IV. An approximation algorithm for the minimum operational cost problem

In this section we first propose a novel optimization framework for the minimum operational cost problem. We then develop a fast approximate solution to the problem based on the optimization framework, through a reduction to a minimum cost multicommodity flow problem whose approximate solution will return a feasible solution to the original problem.

A. An optimization framework

Given a distributed cloud computing environment, to allocate user requests from different web portals to different data centers such that the system throughput is maximized while the operational cost of processing the requests is minimized. An auxiliary flow network $G_f = (V_f, E_f, u_f, c_f)$ at a time slot t is constructed as follows. $V_f = \{DC_i \mid 1 \leq i \leq N\} \cup \{WP_j \mid 1 \leq j \leq M\} \cup \{s_0, t_0\}$ is the set of nodes and s_0 and t_0 are the *virtual* source and destination nodes. $E_f = \{\langle s_0, WP_j\rangle \mid 1 \leq j \leq M\} \cup \{\langle WP_j, DC_i\rangle \mid 1 \leq j \leq M, 1 \leq i \leq N\} \cup \{\langle DC_i, t\rangle \mid 1 \leq i \leq N\}$ is the set of directed edges.

The capacity and cost of each edge in E_f are defined in the following. Associated with each edge $\langle s_0, WP_j\rangle$ for each $WP_j \in \mathcal{WP}$, its capacity $u(s_0, WP_j)$ is the number of requests r_j submitted to web portal WP_j at time slot t, and its cost is zero. Associated each edge $\langle WP_j, DC_i\rangle$ for each $WP_j \in \mathcal{WP}$ and each $DC_i \in \mathcal{DC}$, its capacity is infinity and its cost is the cost sum of network bandwidth consumed between WP_j and DC_i and electricity consumed for processing a request, which is $p_i(t)\alpha_i + p_j^b$. Associated with each edge $\langle DC_i, t\rangle$ for each $DC_i \in \mathcal{DC}$, its cost is zero, and its capacity $u(DC_i, t)$ is the maximum number of requests that can be processed by DC_i at time slot t without violating the average delay D_i of these requests, assuming that the service rate of each server is μ_i. The key here is how to transfer this delay requirement of requests into the edge capacity that corresponds to the processing ability of DC_i. Notice that the number of requests waiting at the queue of DC_i must be limited, as a longer queue will lead to a longer waiting time and the average delay constraint D_i may not be met. To prevent this occurs, the number of requests allocated to DC_i, $\sum_{j=1}^{M} r_{j,i}$, must be upper bounded by the following inequality.

$$\sum_{j=1}^{M} r_{j,i} \leq \lfloor N_i\mu_i - \frac{1}{D_i} \rfloor, \tag{5}$$

where N_i is the number of servers used for processing the requests with the service rate μ_i. The capacity of edge $\langle DC_i, t_0\rangle$ thus is $u(DC_i, t_0) = \lfloor N_i\mu_i - \frac{1}{D_i} \rfloor$ for all i with $1 \leq i \leq N$. Fig. 1 is an illustration of the flow network G_f.

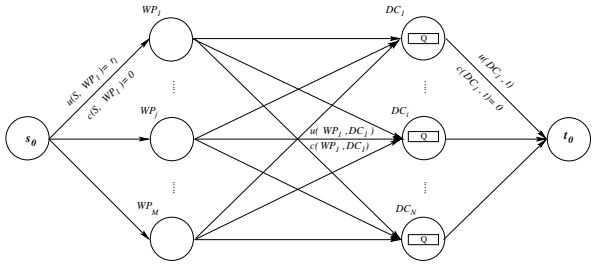


Fig. 1. Flow network $G_f = (V_f, E_f, u, c)$.

B. Algorithm

We now propose a fast approximate solution to the minimum operational cost problem in a distributed cloud computing environment, through a reduction to the minimum cost multicommodity flow problem in G_f, where there are M commodities to be routed from their source nodes WP_j to a common destination node t_0 with demands r_j, represented by a triple $(WP_j, t_0; r_j)$ for all j with $1 \leq j \leq M$ such that the system throughput $\sum_{j=1}^{M} \lambda r_j$ is maximized while the associated operational cost is minimized, where $0 \leq \lambda \leq 1$.

Following Garg and Könemann's optimization framework for the minimum cost multicommodity flow problem [7], the value of λ can be obtained. Meanwhile, a flow f_j with value of $|f_j| = \lambda \cdot r_j$ for each commodity j is obtained too. Since the value $|f_j|$ of flow f_j is no more than $\log_{1+\epsilon}\frac{1+\epsilon}{\delta}$ times the capacity of the most congested edge in G_f, a feasible flow f_j' with value of $|f_j'| = \frac{|f_j|}{\log_{1+\epsilon}\frac{1+\epsilon}{\delta}}$ is then obtained by scaling flow f by $\log_{1+\epsilon}\frac{1+\epsilon}{\delta}$ for all j, $1 \leq j \leq M$. Having the feasible flow $f'(e)$ at each edge $e \in E_f$, the number of servers to be switching on in each DC_i, n_i, can then be determined, which is $n_i = \lceil \frac{\sum_{j=1}^{M} f'(WP_j, DC_i)}{\mu_i} + \frac{1}{\mu_i D_i} \rceil$, while the average delay

[1] $O^*(f(n)) = O(f(n)\log^{O(1)} n)$

D_i of all allocated requests will be met. The minimum cost for the feasible flow f' thus is $\sum_{e \in E_f} c(e) \cdot f'(e)$.

The detailed description of algorithm for the minimum operational cost problem is given in **Algorithm** 1.

Algorithm 1 *Request Allocations at time slot* t

Input: A set of data centers $\{DC_i \mid 1 \leq i \leq N\}$, a set of web portals $\{WP_j \mid 1 \leq j \leq M\}$, request arrival rate r_j at web portal WP_j for each j with $1 \leq j \leq M$, the set $\{p_i(t) \mid 1 \leq i \leq N\}$ of electricity prices for all data centers, the set $\{p_j^b \mid 1 \leq j \leq M\}$ of bandwidth costs, and the accuracy parameter ϵ with $0 < \epsilon \leq 1$.

Output: The minimum operational cost C, the request assignment $\{r_{j,i} \mid 1 \leq j \leq M,\ 1 \leq i \leq N\}$ such that $\sum_{i=1}^{N} \sum_{j=1}^{M} r_{j,i}$ is maximized and the number of servers n_i ($\leq N_i$) in each DC_i to be switched on, where $\sum_{i=1}^{N} r_{j,i} \leq r_j$.

1: Construct an auxiliary flow network G_f, in which there are M commodities in G_f to be routed from their sources to the destination t_0, $\{(WP_j, t_0, r_j) \mid 1 \leq j \leq M\}$;
2: $B \leftarrow \sum_{j=1}^{M} r_j \cdot (\max_{1 \leq i \leq N}\{p_i(t)\}\alpha_i + p_j^b)$;
/* The upper bound on the total cost*/
3: Let f'_j be the feasible flow of each commodity j delivered by applying Garg and Könemann's algorithm for the minimum cost multicommodity flow problem to G_f, and let $f'(e)$ be the fraction of the feasible flow on each edge $e \in E_f$;
4: **for** each web portal $WP_j \in \mathcal{WP}$ **do**
5: **for** each data center $DC_i \in \mathcal{DC}$ **do**
6: $r_{j,i} \leftarrow \lfloor f'(WP_j, DC_i) \rfloor$;
7: **end for**
8: **end for**
9: $n_i \leftarrow \lceil \frac{\sum_{j=1}^{M} f'(WP_j, DC_i)}{\mu_i} + \frac{1}{\mu_i D_i} \rceil$;
/* the number of servers in DC_i to be turned on */
10: $C \leftarrow \sum_{e \in E_f} c(e) \cdot f'(e)$.

C. Algorithm complexity analysis and correctness

In the following we first show that the average delay of any request allocated to data center DC_i is no greater than D_i for all i with $1 \leq i \leq N$. We then analyze the performance and computational complexity of the proposed algorithm as follows.

Lemma 1: For each request from any web portal allocated to a data center DC_i, then the average delay of the request in DC_i is no more than D_i for all i with $1 \leq i \leq N$.

Proof: Following Eq. (5), the capacity $u(e_i)$ for each edge $e_i = \langle DC_i, t \rangle \in E_f$ represents the maximum number of requests that can be served by DC_i while their average delay requirement D_i is met. We show this claim by contradiction. Let $f'(e_i)$ be the fraction of the feasible flow f' on edge e_i, then $f'(e_i) \leq u(e_i) = \lfloor N_i\mu_i - \frac{1}{D_i} \rfloor$ by flow constraints. We now assume that at least one request among the $f'(e_i)$ requests cannot be served by DC_i within the delay D_i, which means the average delay of the requests in DC_i is strictly greater than D_i when all its servers are switched on, i.e., $f'(e_i) > u(e_i) = \lfloor N_i\mu_i - \frac{1}{D_i} \rfloor$. This contradicts the fact that $f'(e_i) \leq u(e_i) = \lfloor N_i\mu_i - \frac{1}{D_i} \rfloor$, the lemma then follows. ■

Theorem 2: Given a distributed cloud computing environment consisting of M web portals and N data centers located in different geographical regions with time-varying electricity prices, there is a fast approximation algorithm for the minimum operational cost problem for fair request rate allocations, which delivers a system throughput no less than $(1-3\epsilon)$ times the optimum while its cost is the minimum. The algorithm takes $O^*(\epsilon^{-2}M^2N^2)$ time, where ϵ is a given constant with $0 < \epsilon \leq 1/3$.

Proof: Since the auxiliary flow network $G_f(V_f, E_f, u, c)$ consists of $|V_f| = M+N+2$ nodes and $|E_f| = MN+M+N$ edges, the construction of G_f takes $O(|V_f| + |E_f|)$ time. Following Theorem 1, Garg and Könemann's algorithm takes $O^*(\epsilon^{-2}M^2N^2)$ time to solve the minimum cost multicommodity flow problem in G_f where there are M commodities to be routed from their sources to a common destination t_0, and the solution obtained (i.e., the system throughput) is $(1-3\epsilon)$ times the optimum. Thus, **Algorithm** 1 takes $O^*(\epsilon^{-2}M^2N^2)$ time and the solution delivered is $(1-3\epsilon)$ times the optimum in terms of the system throughput.

Following the definition of the minimum cost multicommodity flow problem, the solution f' is a feasible solution to the minimum cost multicommodity flow problem, and B is an upper bound on the optimal budget of the problem, thus, the associated cost for the feasible flow f' is the minimum one. In the rest we show that the cost of f' is no more than the optimal cost C^* of the optimal flow f^* for the minimum operational cost problem as follows.

Let $\mathcal{P}^*$ be the union of sets $\mathcal{P}^*(WP_j, t_0)$, where each routing path $p \in \mathcal{P}^*(WP_j, t_0)$ routes part of the demands r_j from WP_j to t_0. and let $\mathcal{P}^*(WP_j, t_0)$ be the set of routing paths of the optimal flow f_j^*, then $\mathcal{P}^* = \cup_{j=1}^{M} \mathcal{P}^*(WP_j, t_0)$. Let f'_j be the feasible flow obtained by **Algorithm** 1. Let $\mathcal{P}^{(1)}(WP_j, t_0)$ be a subset of $\mathcal{P}^*(WP_j, t_0)$ such that the value of the flow of routing paths in it is equal to $|f'_j|$, i.e., there is the same fraction of demands d_j as the feasible flow f'_j routed from WP_j to t_0. Note that one of the routing paths in $\mathcal{P}^{(1)}(WP_j, t_0)$ may need to reduce its flow in order to reach the value $|f'_j|$. We then have

$$\sum_{e \in E_f} f'(e) \cdot c(e) \leq \sum_{j=1}^{M} \sum_{e \in p \in \mathcal{P}^{(1)}(WP_j, t_0)} f^{(1)}(e) * c(e), \quad (6)$$

as the feasible flow f' delivered by **Algorithm** 1 is a minimum cost multicommodity flow, where $|f^{(1)}(e)|$ is the sum of flows in $\cup_{j=1}^{M} \mathcal{P}^{(1)}(WP_j, t_0)$ on edge $e \in E_f$. Meanwhile, $\sum_{j=1}^{M} \sum_{e \in p \in \mathcal{P}^{(1)}(WP_j, t_0)} f^{(1)}(e) \cdot c(e) \leq \sum_{j=1}^{M} \sum_{e \in p \in \mathcal{P}^*(WP_j, t_0)} f^*(e) \cdot c(e) = C^*$, because the left hand side is part of the right hand side. Thus, $\sum_{e \in E_f} f'(e) \cdot c(e) \leq \sum_{j=1}^{M} \sum_{e \in p \in \mathcal{P}^{(1)}(WP_j, t_0)} f^{(1)}(e) * c(e) \leq C^*$, and the cost associated with the feasible flow f' is the minimum. ■

V. Performance Evaluation

In this section we evaluate the performance of the proposed algorithm through experimental simulations, using real-life electricity price data sets.

A. Simulation environment

We consider a cloud environment consisting of five data centers and six web portals, where each data center hosts 15,000 to 25,000 homogeneous servers with identical operating power of 350 $Watts$. The service rate of servers μ_i in different data centers are different, which vary from 2.75 to 3.25, while the bandwidth cost of allocating a request from a web portal to a data center varies from \$0.03 to \$0.06 per hour. The maximum request rates of web portals are from 60,000 to 70,000. Table I summarizes the properties of the data centers and web portals. The accuracy parameter ϵ is set to 0.1. The running time is based on a desktop with 2.66 GHz Intel Core 2 Quad CPU and 8GB RAM. Unless otherwise specified, in the following we adopt this default setting and refer to the proposed algorithm as algorithm `Fair`.

TABLE I
Data centers and web portals

Data Centers	Location	Number of Machines	Service rate (reqs /second)	Operating Power (Watts)
DC_1	Mountain View, CA	15,000	2.75	350
DC_2	Council Bluffs, IA	15,000	2.75	350
DC_3	Boston, MA	20,000	3	350
DC_4	Houston, TX	25,000	3.25	350
DC_5	Lenoir, NC	25,000	3.25	350
Web portals	**Location**	**Maximum request rate**	**Bandwidth cost (\$/hr)**	
WP_1	Seattle	70,000	0.05	
WP_2	San Francisco	70,000	0.04	
WP_3	Dallas	63,000	0.03	
WP_4	Chicago	63,000	0.04	
WP_5	New Mexico	60,000	0.06	
WP_6	Denver	60,000	0.05	

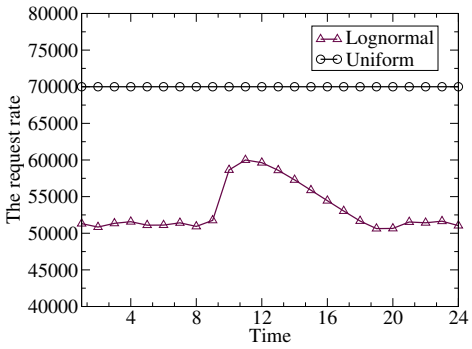


Fig. 2. Two types of request arrival patterns at each web portal.

The request model: In real applications the request rate of a web portal during a day usually starts to rise at around 9:00am, reaching a peak at around 12:00pm and leveling off before 6:00pm. This request arrival pattern is similar to a lognormal process [1] which is referred to *the lognormal request arrival pattern*. However, due to some unexpected social events, a web portal may receive 'burst' requests at a certain time period. These burst requests usually exceed the processing capability of data centers, and this request rate may not change during that time period. We refer to this type of request arrival pattern as *the uniform request arrival pattern*. We will evaluate the performance of the proposed algorithm, using these two different request arrival patterns as illustrated in Fig. 2. In our simulation, notice that the curves of request arrival rates from web portals are shifted according to their time zones.

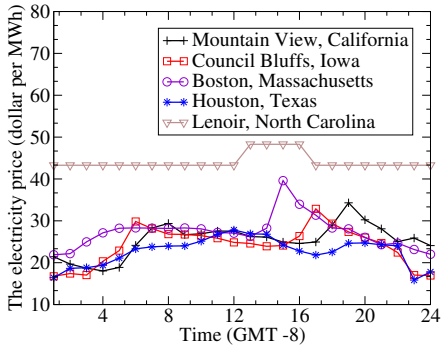


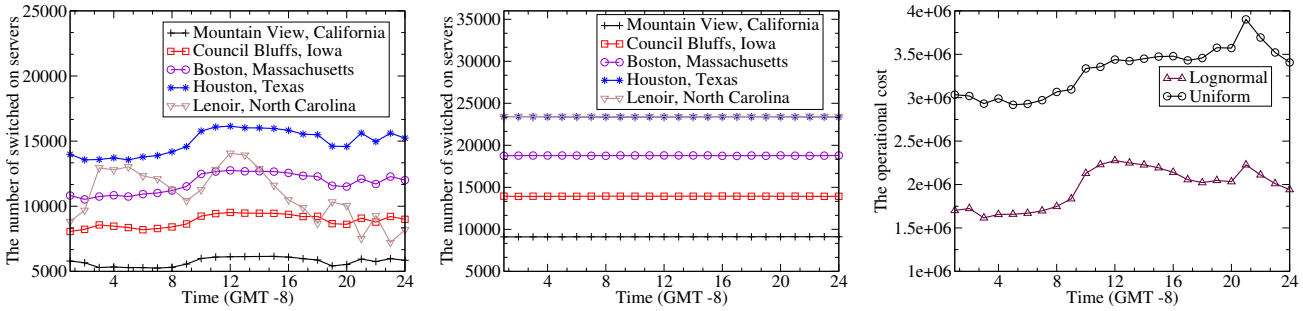
Fig. 3. Electricity prices of different data centers in Table I

Electricity prices for data centers: The average electricity price per hour at each data center is calculated, according to the raw electricity price data obtained from US government agencies [4], [5], and the electricity prices at the data centers are shifted according to their time zones. Fig. 3 depicts the curves of electricity prices of the five data centers listed in Table I.

B. Algorithm performance evaluation

We first investigate the operational cost of the solution delivered by algorithm `Fair`. The average operational costs of algorithm `Fair` during 24 hours are 1,900,490.82 and 3,331,993.98 for the lognormal and uniform request arrival patterns, respectively. To further investigate whether this cost C is optimal, we decrease the budget B in algorithm `Fair` by a certain percentage β of its current cost C where β varies from 0.95 to 0.99. With this new budget $B' = \beta \cdot C$, if algorithm `Fair` is able to deliver another feasible solution while maintaining the current system throughput, then the cost C is not optimal to the problem. Fig. 4 depicts the impact of β on the system throughput, from which it can clearly be seen when the budget B is below C, the system throughput drops below its current level accordingly. Thus, the operational cost C is optimal.

We then evaluate the performance of algorithm `Fair` under both lognormal and uniform request arrival patterns, by incorporating time varying electricity price diversity. Fig. 5 (a)



(a) The number of active servers under the lognormal request arrival pattern

(b) The number of active servers under the uniform request arrival pattern

(c) The hourly operational cost

Fig. 5. The performance evaluation of algorithm `Fair` with time-varying electricity prices.

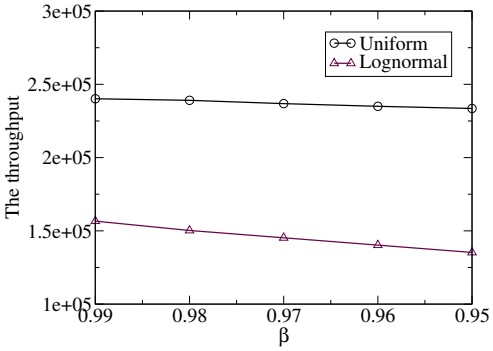


Fig. 4. The impact of the budget $B = \beta \cdot C$ on the system throughput where C is the cost of the solution by `Fair` when the budget $B \geq C$.

shows that under the lognormal request arrival pattern, although the data center located in Lenoir has a nearly identical processing capability as that of the data center in Houston, it contains a fewer servers to be switched on due to the high electricity price there. However, Fig. 5 (b) indicates that under the uniform request arrival pattern, almost all servers in all data centers are to be switched on, as the sum of the number of requests from all web portals is greater than the aggregative processing capability of these data centers. Fig. 5 (c) plots that for both request arrival patterns, the operational costs vary over time. Particularly, the operational costs are much higher from 10am to 6pm, since the electricity prices during this period are normally higher than those in other time periods as shown in Fig. 3.

We thirdly evaluate the system throughput and fairness provided by algorithm `Fair` by varying the accuracy parameter ϵ from 0.05 to 0.1. Notice that with this setting, all requests under the lognormal request arrival pattern can be processed immediately by the data centers as their aggregative processing capability is much larger than the accumulative request load from all web portals. The rest thus focuses only on the performance of algorithm `Fair` under the uniform request arrival pattern. To investigate how far the system throughput delivered by algorithm `Fair` from the optimal one OPT, we use the maximum flow in G_f from s_0 to t_0 as an upper bound on OPT. It must be mentioned that this upper bound estimation is very conservative because it does not consider the budget constraint. Fig. 6 (a) plots the curve of the system throughput while Fig. 6 (b) depicts the throughput approximation ratio curve by algorithm `Fair`. From Fig. 6 (b) it can be seen that the throughput ratio by algorithm `Fair` is no less than 0.96 when $\epsilon = 0.05$, and this value decreases with the increase of ϵ. With different accuracy values ϵ, the running time of algorithm `Fair` is different too, which is illustrated in Fig. 6 (c). Obviously, a larger ϵ will result in a shorter running time. Fig. 7 depicts the percentage of numbers of requests from each web portal WP_j allocated to data centers, $\lambda_j = \sum_{i=1}^{N} r_{j,i}/r_j$ by algorithm `Fair` under the uniform request arrival patter. It can be seen that the solution delivered by the algorithm maintains the request rate fairness among the web portals.

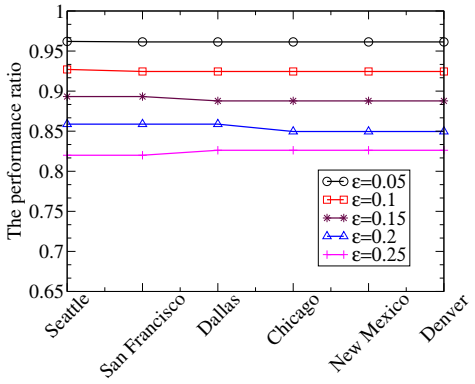
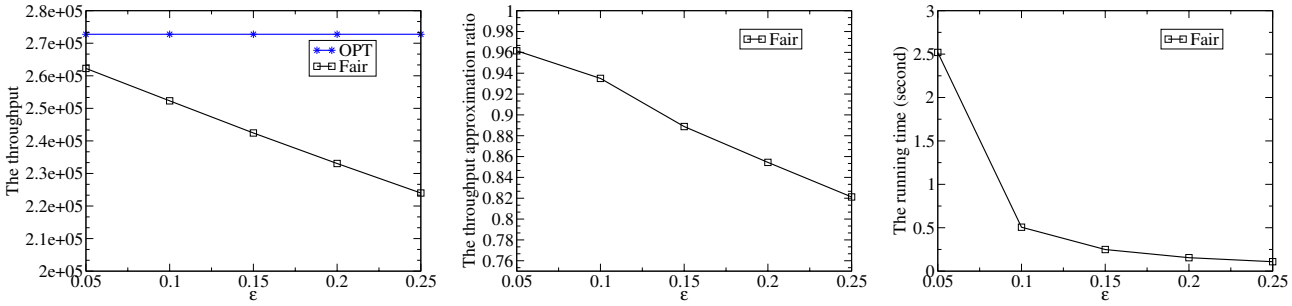


Fig. 7. The percentage of numbers of requests from each web portal allocated.

We finally study the scalability of algorithm `Fair` by assuming that there is a 'virtual' large scale cloud service provider consisting of from 10 to 20 data centers randomly deployed in some of 48 potential states in the States and one data center is deployed in each chosen state. We evaluate the scalability of algorithm `Fair` by varying the number of data centers from 10 to 20 while keeping the request rate of each



(a) The system throughput (b) The throughput approximation ratio (c) The running time

Fig. 6. The performance of algorithm `Fair` under the uniform request arrival pattern with different accuracy values ϵ.

web portal unchanged. We set ϵ to be 0.1. The electricity price for each data center is randomly selected from a prior given set of electricity prices. The time zone of each data center is randomly chosen from GMT-8 to GMT-5. Fig. 8 indicates that algorithm `Fair` takes at most 8 seconds and 4 seconds to find a solution to allocate all coming requests during 24 hours under both uniform and lognormal request arrival patterns, respectively.

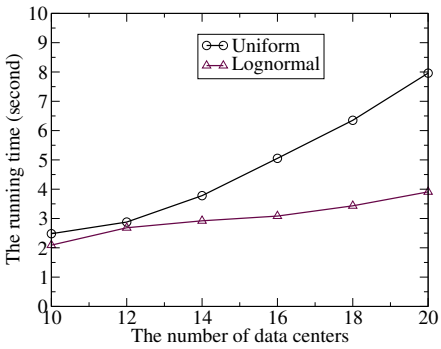


Fig. 8. The scalability of algorithm `Fair` under both lognormal and uniform request arrival patterns.

VI. Conclusion

In this paper, we considered the minimum operational cost problem for fair request rate allocations in a distributed cloud service environment, by incorporating time-varying electricity prices and request rates, for which we first proposed an optimization framework. We then developed a fast approximation algorithm with a provable approximation ratio. We finally conducted extensive experiments by simulation to evaluate the performance of the proposed algorithm, using real-life electricity price data sets. The experimental results demonstrate that the proposed algorithm is very promising.

References

[1] T. Benson, A. Anand, A. Akella, and M. Zhang. Understanding data center traffic characteristics. *Proc. of Sigcomm Workshop: Research on Enterprise Networks*, ACM, 2010.

[2] N. Buchbinder, N. Jain, and I. Menache. Online job-migration for reducing the electricity bill in the cloud. *Proc. of Networking'11*, IFIP, 2011.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. 3rd Edition, MIT Press, 2009.

[4] U.S. energy information administration (EIA). *http://www.eia.doe.gov/*.

[5] Federal Energy Regulatory Commission. *http://www.ferc.gov/*.

[6] Y. Guo, Z. Ding, Y. Fang, and D. Wu. Cutting down electricity cost in Internet data centers by using energy storage. *Proc of Globecom'11*, IEEE, 2011.

[7] N. Garg and J. Könemann. Faster and simpler algorithms for multi-commodity flow and other fractional packing problems. *Proc of FOCS'98*, IEEE, 1998.

[8] Google Data Centers. *http://www.google.com/about/datacenters/*.

[9] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris. *Fundamentals of Queueing Theory*. 4th Edition, Wiley Press, 2008.

[10] IBM cloud computing. *http://www.ibm.com/cloud-computing/us/en/*.

[11] J. Kaplan, W. Forrest, and N. Kindler. Revolutionizing data centre energy efficiency. *McKinsey*, Technique Report, 2008.

[12] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T. D. Nguyen. Managing the cost, energy consumption, and carbon footprint of Internet services. *Proc. of SIGMETRICS'10*, ACM, 2010.

[13] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew. Geographical load balancing with renewables. *Performance Evaluation Review*, ACM, Vol. 39, No. 3, pp.62–66, 2011.

[14] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew. Greening geographical load balancing. *Proc. of SIGMETRICS'11*, ACM, 2011.

[15] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for Internet-scale systems. *Proc. of SIGCOMM'09*, ACM, 2009

[16] L. Rao, X. Liu, M. D. Ilic, and J. Liu. Distributed coordination of Internet data centers under multiregional electricity markets. *Proc. of IEEE*, IEEE, 2011.

[17] L. Rao, X. Liu, L. Xie, and W. Liu. Minimizing electricity cost: optimization of distributed Internet data centers in a multi-electricity-market environment. *Proc. of INFOCOM'10*, IEEE, 2010.

[18] L. Rao, X. Liu, L. Xie, and W. Liu. Coordinated energy cost management of distributed Internet data centers in smart grid. *IEEE Transactions on Smart Grid*, IEEE, Vol. 3, No. 1, pp.50–58, 2012.

[19] A. Riska, N. Mi, E. Smirni, and G. Casale. Feasibility regions: exploiting tradeoffs between power and performance in disk drives. *Performance Evaluation Review*, ACM, Vol. 37, No. 3, pp.43-48, 2009.

[20] Y. Zhang, Y. Wang, and X. Wang. GreenWare: greening cloud-scale data centers to maximize the use of renewable energy. *Proc. of MIDDLEWARE'11*, ACM, 2011.

[21] Z. Zhang, M. Zhang, A. Greenberg, Y. C. Hu, R. Mahajan, and B. Christian. Optimizing cost and performance in online service provider networks. *Proc. of NSDI'10*, ACM 2010.