DAHM: A Green and Dynamic Web Application Hosting Manager across Geographically Distributed Data Centers

ZAHRA ABBASI, TRIDIB MUKHERJEE, GEORGIOS VARSAMOPOULOS, and SANDEEP K. S. GUPTA, Arizona State University, Tempe, AZ

Dynamic Application Hosting Management (DAHM) is proposed for geographically distributed data centers, which decides on the number of active servers and on the workload share of each data center. DAHM achieves cost-efficient application hosting by taking into account: (i) the spatio-temporal variation of energy cost, (ii) the data center computing and cooling energy efficiency, (iii) the live migration cost, and (iv) any SLA violations due to migration overhead or network delay. DAHM is modeled as *fixed-charge min-cost flow* and *mixed integer programming* for stateless and stateful applications, respectively, and it is shown NP-hard. We also develop heuristic algorithms and prove, when applications are stateless and servers have an identical power consumption model, that the approximation ratio on the minimum total cost is bounded by the number of data centers. Further, the heuristics are evaluated in a simulation study using realistic parameter data; compared to a performance-oriented application assignment, that is, hosting at the data center with the least delay, the potential cost savings of DAHM reaches 33%. The savings come from reducing the total number of active servers as well as leveraging the cost efficiency of data centers. Through the simulation study, the article further explores how relaxing the delay requirement for a small fraction of users can increase the cost savings of DAHM.

Categories and Subject Descriptors: D.4.7 [**Operating Systems**]: Organization and Design—*Distributed systems; hierarchical design; interactive systems*; D.4.8 [**Operating Systems**]: Performance; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Distributed systems; performance evaluation*

General Terms: Design, Management, Performance

Additional Key Words and Phrases: Data center cost efficiency, server management, workload management, hosting management

ACM Reference Format:
Abbasi, Z., Mukherjee, T., Varsamopoulos, G., and Gupta, S. K. S. 2012. DAHM: A green and dynamic web application hosting manager across geographically distributed data centers. ACM J. Emerg. Technol. Comput. Syst. 8, 4, Article 34 (October 2012), 22 pages.
DOI = 10.1145/2367736.2367745 http://doi.acm.org/10.1145/2367736.2367745

T. Mukherjee is currently affiliated with Xerox Research Center.

This article extends the conference version titled “Dynamic hosting management of Web based applications over clouds,” which will appear in HiPC 2011. We have (i) revised the entire modeling under more realistic assumptions for workload management and migration cost, (ii) developed NP-hardness proof of the DAHM problem, as well as performance bound guarantee for the Greedy algorithm, (iii) revised related work and simulation study.

This work has been partly funded by NSF CRI grant no. 0855527, CNS grant no. 0834797 and Intel Crop.

Authors’ addresses: Z. Abbasi (corresponding author), Arixona State University, Tempe, AZ; email: Zahra.abbasi@asu.edu; T. Mukherjee, Xerox Research Center, India; G. Varsamopoulos, S. K. S. Gupta, Arizona State University, Tempe, AZ.

DOI 10.1145/2367736.2367745 http://doi.acm.org/10.1145/2367736.2367745

1. INTRODUCTION

With the increasing prevalence of Internet-based computing services such as online gaming [Beskow et al. 2009], cloud-based services [Kumar and Lu 2010], and search engines, the energy consumption in data centers to host such services has skyrocketed, and so has the importance of saving energy in data centers. According to a report by Intel and Microsoft [Koomey et al. 2009], the energy cost accounts for over 10% of the Total Cost of Ownership (TCO) of a data center.

Data center energy cost can be lowered through cloud computing as it facilitates a dynamic, demand-driven allocation of computation load across data centers. Applications can be assigned to virtual machines (VMs) independent of the physical infrastructure. Virtualization provides a cloud flexibility to host an application on the most cost-efficient data center at the time. However, such cost-efficient application hosting needs to be aware of data centers' cost efficiency metrics, the live migration cost for stateful applications, applications' performance requirements, and their workload variations.

The cost efficiency of a data center depends on both the *energy efficiency*, for example, its Power Usage Effectiveness (PUE) and the power proportionality of its servers as well as on the *cost model*, for example, the cost model of the spatio-temporal variation in the electricity price [Gupta et al. 2011b; Rao et al. 2010b]. Figure 1 shows the temporal variation in the electricity price for three U.S. cities. PUE, which measures the efficiency of a cooling system and any source of power consumption other than the computing equipment in data centers, may vary among data centers as well [FEMP 2010]. Moreover, the computing power consumption model of data centers (power consumption over the utilization of the servers) may vary depending on the power behavior of the servers. The energy efficiency of data centers improves with the use of power-proportional servers [Krioukov et al. 2010; Varsamopoulos and Gupta 2010; Chun et al. 2010]. Servers should be *ideally power-proportional*, that is, they should consume zero power when idle and have a linear increase of power consumption with respect to utilization [Barroso and Hölzle 2007]. However, in reality, servers do not have zero power at idle [Varsamopoulos et al. 2010; Hsu and Poole 2011].

Further, dynamically hosting applications in data centers should be aware of the network delay and bandwidth overhead during migration (e.g., user state data). This overhead depends on the type of Web applications, which can be either stateless or stateful. In *stateless applications*, for example, search engines, the state of online users is not recorded; whereas *stateful applications*, for example, multiplayer online games, keep track of the state of users [Beskow et al. 2009]. Therefore, stateful applications tend to induce higher migration cost. All related work approaches are designed for stateless applications, where migration costs are negligible.

Finally, for Web applications, requests may originate from different locations or geographical areas. As such, the network delay from these locations to the hosting data centers might also impact the end-to-end delay of the application users. This may prevent an application from being hosted at certain locations. The population of online users for a specific Web application varies over time and has a cyclic behavior (i.e., hourly, daily, weekly, etc.). For instance, Figure 2, demonstrating the online users for an entertainment application, shows that the peak traffic time for different locations does not occur at the same time. All the aforementioned spatio-temporal variables make some data centers the most cost efficient at one time, and other data centers at another time.

In this article, we propose *Dynamic Application Hosting Management* (DAHM) to perform server and workload management across data centers. DAHM uses the flexibility of cloud computing to decrease the cost of hosted applications by considering: (i) the power consumption behavior and the energy cost of data centers within a cloud, and

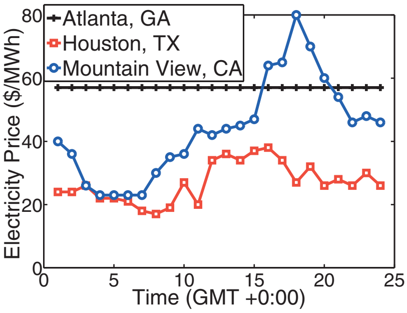

Fig. 1. Hourly electricity price data for three major locations of Google IDCs on May 2nd, 2009 (data source Rao et al. [2010b]).

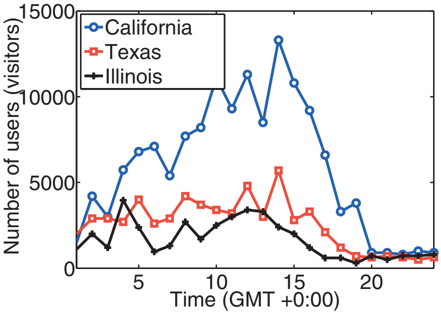

Fig. 2. Hourly number of online users from three states for an entertainment Web site hosted at GoDaddy.com.

(ii) the applications' performance requirement and bandwidth cost of live migration. The spatial and temporal variation of electricity cost and data centers' average performance have been previously considered in workload distribution algorithms [Qureshi et al. 2009; Le et al. 2010]. However, our model improves the cost savings over the previous approaches under nonpower-proportional servers. Workload distribution management reduces mainly the nonidle power consumption, whereas server management (i.e., resizing the active server set) reduces the idle power as well. Server and workload management is also proposed in some recent works [Rao et al. 2010b; Liu et al. 2011]. However, to the best of our knowledge, a cost-optimal solution to the workload and server management across data centers has not been proposed yet, nor has a study on the approximation ratio of polynomial-time heuristics been performed. We model the workload and server management as a Mixed Integer Programing (MIP) and show that the Linear Programming (LP) relaxation of the problem does not always provide an optimal-tight solution. To this end, the following contributions have been made.

—*In formulating the DAHM problem.* DAHM is formulated as a *Mixed Integer Programming (MIP)*, and is proven NP-hard (Section 3). The objective function is a cost model that accounts for energy cost of data centers and migration cost, where the constraints are set to respect the performance goals (i.e., delay). In the case of stateless applications, DAHM is shown to be a specific type of MIP that is a Fixed-Charge Min-Cost Flow (FCMCF) problem.

—*In designing solutions to the problem.* Optimal solutions for DAHM in both stateless and stateful applications, referred as zero migration cost and nonzero migration cost cases, respectively, are provided by use of branch-and-bound, whose worst-case time complexity is exponential with respect to the product of the number of areas and the number of data centers. Further, polynomial-time greedy algorithms are developed that dynamically decide on the number of active servers and the workload share (Section 4). The greedy algorithm for the case of zero migration cost is shown to be an $|S|$-approximation ($|S|$ is number of data centers) when the servers have a homogeneous power consumption model, which further improves to a 2-approximation when, additionally, workload can be split under no network delay constraint.

Our simulation study using realistic workloads, network delay, electricity price, and data center power consumption model shows that using DAHM heuristic solutions, the relative cost can be saved up to 33% depending on the data center power consumption model and the heterogeneity of data centers. The results also show that the total electricity cost of the DAHM optimal solution decreases even with nonpower-proportional servers with respect to a reference performance-oriented static assignment, a result that contradicts previously published work [Le et al. 2010; Qureshi et al. 2009]. Also a simulation study is performed to explore the cost efficiency of DAHM when allowing to trade off on QoS violations. The results show that by relaxing the performance requirements for a small fraction of users, which incurs a cost, the total cost efficiency of DAHM can be further improved

The rest of the article is organized as follows. Section 2 presents the related work. Section 3 presents the system model including the cost model, the performance model, and the DAHM problem formulation followed by the online algorithms for DAHM in Section 4. Section 5 presents the simulation-based evaluation of DAHM. Further, Section 6 concludes the work. Finally, Appendix A provides a theoretical proof for performance bound guarantee of the greedy algorithm.

2. RELATED WORK

Most previous related work focuses on improving the energy (cost) efficiency through workload and server management of one data center [Lin et al. 2011; Chase et al. 2001; Kusic et al. 2009; Abbasi et al. 2012; Krioukov et al. 2010; Chen et al. 2005]. However, virtualization and the spatio-temporal variation of electricity price offer leveraging opportunities to perform cost-efficient workload placement across data centers [Rao et al. 2010a, 2010b; Le et al. 2010; Qureshi et al. 2009; Liu et al. 2011].

Qureshi et al. used heuristics to quantify the potential economic gain of considering electricity price in the location of computation [Qureshi et al. 2009]. Through simulation using realistic historical electricity price and real workload, they reported that judicious location of computation load may save millions of dollars on the total operation cost of data centers. They also showed that the magnitude of cost savings depends on how power-proportional the servers are and whether there is a constraint on the network bandwidth. They found that the cost saving is the highest when servers are ideally power-proportional and when the available network bandwidth is unconstrained.

Rao et al. considered the load distribution across data centers with the objective of minimizing current energy cost subject to delay constraints [Rao et al. 2010b]. The energy cost considered accounted for the average energy cost of active servers (i.e., active servers are assumed to operate at an average utilization and frequency). The authors used linear programming techniques and min-cost flow model to find an approximate solution. We additionally enhance the power consumption model of active servers to be dependent on their current utilization and provide theoretical and numerical analysis of the approximation compared to the optimal solution. Rao et al.

extended their preceding scheme by developing a joint optimization of server management (i.e., resizing the active server set) and power management (i.e., CPU dynamic voltage and frequency scaling) across data centers using *general Benders decomposition* [Rao et al. 2010b]. Further, Le et al. [2010] developed a workload scheduling across data centers where the problem is modeled as a nonlinear optimization problem and it is solved using *simulated annealing*. Their simulation results showed that by leveraging the electricity price, significant cost can be saved when servers are ideally power-proportional, and the cost saving decreases when servers have greater-than-zero idle power. Liu et al. derived two distributed algorithms for achieving optimal geographical load balancing [Liu et al. 2011]. Their optimal algorithm is based on relaxing the number of active servers from an integer to a continuous variable, however, we show that this relaxation does not always incur a tight bound compared to the optimal solution.

Finally, Buchbinder et al. developed a scheme for online job migration across data centers to reduce the electricity bill [Buchbinder et al. 2011]. They proved a competitive bound of $\log(n)$ for their proposed algorithm, where n is the total number of servers across the cloud. However, due to the complexity of the algorithm, a heuristic easy-to-implement online algorithm is proposed which is evaluated through simulations using real electricity pricing and job workload data. The assumptions to derive the analytical bound were more suited to batch jobs. However, we model dynamic hosting of Web-based applications, where migration cost is a function of the number of migrated user connections. We show that the idle power consumption is a significant adverse factor on the cost efficiency of the optimal solution, because, contrary to the previous related work, we allow servers to be utilized at any level.

The DAHM scheme developed in this article complements the aforementioned research in the following ways. First, the applications' migration cost is incorporated into the optimization problem, and it can be a significant adverse factor for stateful applications. Second, all related work shows the significant impact of the servers' idle power on the cost saving of workload distribution across data centers, which suggests that server management is needed to reduce the effect of idle power. Although previous works [Rao et al. 2010b; Liu et al. 2011] considered server provisioning, their solutions modeled the number of servers as a real. We show results when the number of servers is constrained to be an integer. We also perform study to show how time zone difference over areas and electricity price variation can be leveraged to minimize cost, while incurring negligible performance violation for a small fraction of the population. In the next section, we present the system model and formally define the DAHM problem.

3. SYSTEM MODEL

We assume a network flow optimization model on a bipartite graph (see Figure 3). End users' requests arrive from $|A|$ geographically distributed front-ends (i.e., the sources) where $A = \{a_1 \dots a_j \dots a_{|A|}\}$ denotes the set of front-ends (we use the term area and front-end interchangeably in the rest of the work). The geographical front-ends may be network prefixes, or even geographic groupings (states and cities). The workload must be distributed among the $|S_t|$ available data centers in the cloud (i.e., sink), where $S_t = \{s_{1,t} \dots s_{i,t} \dots s_{|S|,t}\}$ denotes the set of available data centers in the cloud, and each $s_{i,t}$ represents the number of available servers in data center i at time t. The optimization is a two-level process: (i) deriving the number of required active servers $y_{i,t}$ ($y_{i,t} \in \mathbb{N}_0,\ 0 \leqslant y_{i,t} \leqslant s_{i,t}$) at each data center i, and (ii) deriving the traffic fractions $x_{i,j,t}$ ($x_{i,j,t} \in \mathbb{R},\ 0 \leqslant x_{i,j,t} \leqslant 1$) from each area j to each data center i. There are constraints (server capacity, Service Level Agreement (SLA), etc.) and the optimization must reconcile a number of competing objectives (server electricity costs,

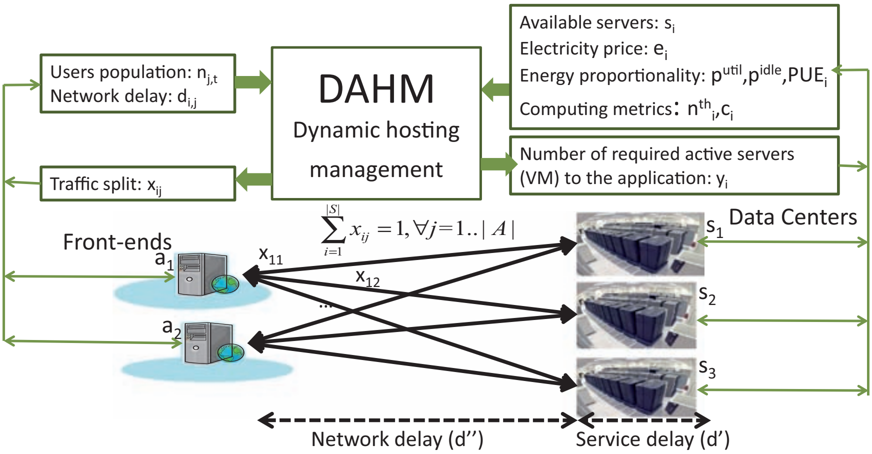
Users population: $n_{j,t}$
Network delay: $d_{i,j}$
DAHM
Dynamic hosting management
Available servers: s_i
Electricity price: e_i
Energy proportionality: p^{util},p^{idle},PUE_i
Computing metrics: n^{th}_i,c_i
Number of required active servers (VM) to the application: y_i
Traffic split: x_{ij}
$\sum_{i=1}^{|S|} x_{ij} = 1, \forall j = 1..|A|$
Front-ends
a_1
a_2
x_{11}
x_{12}
...
s_1
Data Centers
s_2
s_3
Network delay (d″)
Service delay (d′)

Fig. 3. DAHM system model.

Table I. Symbols and Definitions

Symbol	Definition
t	epoch index
i	index of data centers
j	index of areas
$x_{i,j,t}$	workload share of area j to DC i
$y_{i,t}$	number of active servers at DC i
τ	length of epochs (in second)
p_i^{idle}	server idle power at DC i
p^{util}	server peak power minus p^{idle}
c_i	avg. util. of a user on a server at DC i
u_i^{th}	threshold util. of servers at DC i
n_i^{th}	affordable no. of users for servers
s_i	number of available servers at DC i

Symbol	Definition
d	total delay of a request
d'	data center delay
d''	network delay between areas and DCs
d^{ref}	total reference delay
d'^{ref}	service reference delay
e_i	electricity cost at DC i
β	migration cost per migration
α	switching cost of a new server
η	performance violation cost per each user
$n_{j,t}$	avg. number of online users in area j
si	% of new users over an epoch
so	% of users to sign out over an epoch

migration costs, client performance, etc.). We also assume that the optimization takes place in a centralized location, such that information about the system is collected at a single point, the optimal number of servers and traffic splits are derived, and then the solutions are passed to the data centers and front-ends. The optimization is performed regularly with a frequency that is fast enough to capture the electricity cost variation yet slow enough to prevent computation and network overhead (in our simulation, we set the decision interval to one hour). The optimal solutions are derived by some assumed prediction mechanism, slightly before the time step, namely "epoch", starts.

3.1. Performance Modeling

Internet applications are usually delay sensitive such that their Quality-of-Service (QoS) mainly depends on the end-to-end *delay*. For that we assume that for the end users to experience a high QoS, their delay should not go above a *reference delay*, d^{ref}, as tolerated by the application. The delay d experienced by a user consists of the service delay d', that is, data center delay, and the network delay d'', that is, the delay between the front-end and the data center; the total delay becomes $d = d' + d''$.

In Internet data centers, the SLA statistically bounds the delay, that is, the delay of q percent of requests should not go beyond the reference service delay d^{ref}. We use the performance model by Chase et al. to guarantee the SLA for data centers [Abbasi et al.

2010; Chase et al. 2001]. This model asserts that the service delay strongly depends on the servers' utilization level; specifically, the SLA is guaranteed under a utilization threshold. This threshold depends on the capacity of the servers and the type of the application. Specifically, there is a threshold u_i^{th} associated with each data center's servers, such that, if servers are not utilized above that point, the service delay d_i' respects the SLA, that is, $d_i' \leqslant d'^{\text{ref}}$ for q percentage of users.

The aforesaid utilization-to-delay model can be replaced by queuing models, for example, M/M/n or GI/G/n, mainly because the average delay is linearly correlated to the arrival rate λ and the per-request utilization u ($d' = f(\frac{\lambda}{\text{service-time}})$, where $u = \frac{\lambda}{\text{service-time}}$), based on Little's Law. In GI/G/n, the coefficients of variation of workload arrival rate and of service time come into play but they do not change the nature of the problem.

In modeling the network delay, we consider the delay differs depending on the network distance between areas and data centers. Also this delay may vary over time, depending on the network congestion. This is why we denote a delay as $d_{i,j,t}$ to mark the dependence on data center i, the front-end j, and the epoch t. Hence, $d_{i,j,t} = d'_{i,t} + d''_{i,j,t}$.

3.2. Workload Modeling

Let N_t be set of average numbers of online users in the areas for an epoch t, where $N_t = \{n_{1,t} \ldots n_{j,t} \ldots n_{|A|,t}\}$. The set N_t varies over time because, first, different applications have different local peak times during a day; second, the traffic peaks across the areas differ due to the time zone difference. We assume the population and distribution of the users and the electricity price to vary over time and space. However, we assume that these values remain constant within each epoch.

3.3. Energy Costs

We assume that the energy cost of a data center hosting a Web application is a function of the power model of its servers, its cooling energy, and the electricity price.

We model the power consumption of a server at data center i at epoch t as: $p_{i,t} = u_{i,t} p_i^{util} + p_i^{idle}$, where p_i^{idle} is the per-server average idle power consumption for that data center, p_i^{util} is the additional power consumption of a server at full utilization with respect to idle, and $u_{i,t}$ is the utilization of the server at epoch t.

The utilization of a server depends on its workload and its physical characteristics. The workload of a server is a function of its online users. Therefore, we assume the following linear model for utilization of a server: $u_{i,t} = c_i n_{i,t}$, where c_i is the average utilization that one online user imposes on the server, and $n_{i,t} = \sum_{j=1}^{|A|} x_{i,j,t} n_{j,t}$ is the total number of users that are assigned to the data center i at epoch t. This model is frequently used in existing literature and experimental results show its sufficiency [Abbasi et al. 2010; Chase et al. 2001].

Usually, many servers are allocated to the application. Assume n_i^{th} to be the total number of users that a single server in data center i can afford, that is, $c_i n_i^{th} = u_i^{th}$, then the data center's cumulative idle power consumption equals $y_{i,t} p_i^{idle}$, where $y_{i,t}$ is the number of active servers, calculated as $\lceil n_{i,t}/n_i^{th} \rceil$. Turning servers on for the application also incurs a power cost [Kusic et al. 2009; Lin et al. 2011]; for that we consider the constant cost of α_i that is incurred whenever a server is turned on at data center i. As we will see shortly, we assume a server provisioning scheme that turns off servers which are not assigned to the application.

A data center's total power equals the sum of computing and noncomputing equipment power consumption (e.g., cooling power), and can be estimated as the product of its PUE and computing power. There are many data center metrics that evaluate its overall energy efficiency with respect to its computing energy efficiency. We choose PUE since it captures the data center energy inefficiency of the noncomputing equipment with respect to computing energy in a linear way. To calculate the total energy cost of a data center in an epoch, it suffices to multiply its total power consumption into the electricity price. We denote the electricity price by $e_{i,t}$, thus the total energy cost of an application hosted in data center i during epoch t equals

$$Cost_{i,t}^{energy} = \left(\sum_{j=1}^{|A|} x_{i,j,t} n_{j,t} c_i p_{i,t}^{util} + y_{i,t} p_i^{idle} \right) PUE_i e_{i,t} \tau + \sum_{i=1}^{|S|} \alpha_i (y_{i,t} - y_{i,t-1})^+, \tag{1}$$

where the "+" indicates that only when the difference is positive is the cost considered, otherwise there is no cost.

3.4. Migration Cost

Dynamic workload distribution for stateful applications may require live migration (i.e., online users' state information should migrate from source to destination data center). Migration imposes a cost in terms of increase in network bandwidth consumption, and delaying the service for affected online users. Therefore, we consider a uniform, per-user migration cost β, assuming equal-sized state information for all users. The calculation of the migration cost is based on the number of online users who have been migrated, as follows. Eq. (1) suggests that if a front-end assignment to a data center between two intervals changes, then migration is performed. Therefore, we can calculate the number of migrated users for each data center and front-end by calculating the difference in the number of assigned users between two consecutive epochs. However, we choose not to directly take the difference between the previous epoch's $(t-1)$ assignment and the next epoch's (t) assignment, that is, $n_{j,t}x_{i,j,t} - n_{j,t-1}x_{i,j,t-1}$, because we have to account for the users that are signing out in epoch $t-1$ (and therefore their connections are not migrated) and the users that are signing in, in epoch t (and therefore their connections did not exist at migration time). Let si denote the average fraction $(0 \leqslant si \leqslant 1)$ of new users out of the total users at each area over epochs, and so denote the average fraction $(0 \leqslant so \leqslant 1)$ of users at each area who sign out during each epoch, then the migration cost for a data center i at time t can be formulated as

$$\text{cost}_{i,t}^{\text{migration}} = \beta \sum_{j=1}^{|A|} \big((1 - si) n_{j,t} x_{i,j,t} - (1 - so) n_{j,t-1} x_{i,j,t-1} \big)^+. \tag{2}$$

Each of the si and so parameters can be estimated from the other based on preservation of flow, expressed by this relation: $n_{j,t}(1 - si) = n_{j,t-1}(1 - so)$ (i.e., the users that did not sign out in epoch $t-1$ should be equal to the online users that did not just sign in, in the epoch t). The decision to migrate workload is justified by the expectation that it can complete its execution by posing lower energy cost on another data center. The migration depends on two parameters: (i) the longevity of user connection; naturally, it is rarely beneficial to migrate a short running job as the benefit does not outweigh the migration costs; and (ii) the migration cost; if the migration cost is much higher than the difference between energy cost efficiency of two data centers for processing an online user workload, the migration never happens. If the migration cost is much lower than the difference between energy cost efficiency of two data centers, it always happens. In our simulation study (Section 5), we assume long connections for online users, and investigate the migration cost impact on the DAHM cost saving with respect to the average energy-cost benefit of migration. We refer DAHM problem as the DAHM for zero migration cost case, and the DAHM for nonzero migration cost case problem in the rest of article, where the former assumes $\beta = 0$ (i.e., stateless applications), and the latter assumes $\beta \neq 0$ (i.e., stateful applications).

Minimize

$$\begin{aligned}\text{Cost} &= \text{Cost}^{\text{energy}} + \text{Cost}^{\text{migration}}\\ &= \sum_{t=1}^{T}\left(\sum_{i=1}^{|S_t|}\left(\sum_{j=1}^{|A|} x_{i,j,t} n_{j,t} c_i p_{i,t}^{util} + y_{i,t} p_i^{idle}\right)\text{PUE}_i e_{i,t}\tau + \sum_{i=1}^{|S_t|}\alpha_i (y_{i,t} - y_{i,t-1})^{+}\right.\\ &\left.+\beta\sum_{i=1}^{|S_t|}\sum_{j=1}^{|A|}\left((1-si)n_{j,t}x_{i,j,t} - (1-so)n_{j,t-1}x_{i,j,t-1}\right)^{+}\right)\end{aligned} \tag{3}$$

subject to

$$(\text{Service constraint})\ \forall i,j,t:\ \ 0 \leqslant x_{i,j,t} \leqslant 1 \text{ and, } \sum_{i=1}^{|S_t|} x_{i,j,t} = 1, \tag{4}$$

$$(\text{Idle power constraint})\ \forall i,j,t:\ \ y_{i,t} \in \mathbb{N}_0 \text{ and, } y_{i,t} \geq \frac{\sum_{j=1}^{|A|} x_{i,j,t} n_{j,t}}{n_i^{th}}, \tag{5}$$

$$(\text{Capacity constraint})\ \forall i,t:\ \ 0 \leqslant y_{i,t} \leqslant s_{i,t}, \tag{6}$$

$$(\text{Performance constraint})\ \forall i,j,t:\ \ d_{i,j,t} = d_i^{\prime\text{ref}} + d''_{i,j,t} \text{ and, } (d^{\text{ref}} - d_{i,j,t})x_{i,j,t} \geq 0. \tag{7}$$

Fig. 4. Mixed Integer Programming (MIP) formulation of the Dynamic Application Hosting Management (DAHM) problem.

3.5. DAHM Problem Formulation

The problem can be summarized as follows.

> *DAHM problem.* Given an application with a specific delay requirement d^{ref}, a cloud S_t in which the application can be hosted in a dynamic way, a spatio-temporal variation of the electricity price, $e_{i,t}$, a spatio-temporal variation of the number of the online users N_t, find the hosting for each epoch t that minimizes the sum of energy and migration cost, Eq. (1) and Eq. (2).

All aforementioned costs are assumed to be monetary. We can model the application hosting problem as an optimization problem where the objective is minimizing the total cost as shown in Figure 4. Cost minimization is subject to the following constraints.

—*Service constraint* (Eq. (4)), asserts that all users of every area should be assigned to a data center, and that there are no double assignments in either direction.

—*Idle power constraint* (Eq. (5)), makes sure that the idle power consumption of all active servers is accounted.

—*Capacity constraint* (Eq. (6)), that is, the number of assigned active servers to the application in a data center should not exceed the available servers (denoted by $s_{i,t}$) in that data center.

—*Performance constraint* (Eq. (7)), that is, the traffic of end users should be split among data centers whose network and service delay is less than the users' delay requirement.

A solution to this problem would specify, at each epoch, how many servers in each data center should be assigned to the application (i.e., $y_{i,t}$) and what portion of each area's traffic should be assigned to which data center (i.e., $x_{i,j,t}$). Observe that some of the variables are reals (i.e., $x_{i,j,t}$) and some are integers (i.e., $y_{i,t}$). Therefore, due to linearity of all equations (both the objective function and the constraints), the problem is a Mixed Integer Programming, (MIP). MIP is a well-known and general NP-hard problem class for which generic solutions have high computational complexity. For the case of zero migration cost, DAHM can be formulated as a more specific problem that is a Fixed Charge Min-Cost Flow (FCMCF). FCMCF is also NP-hard

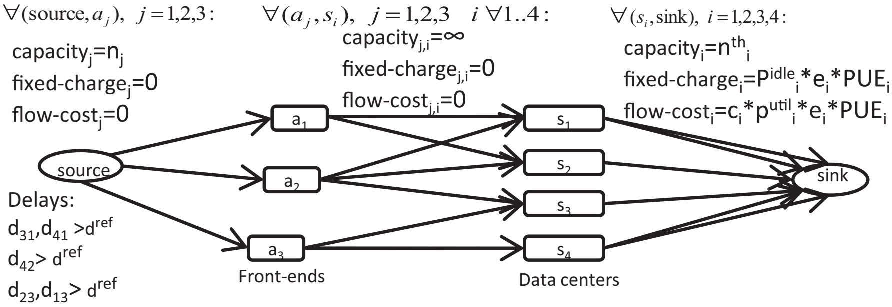
$\forall(\text{source}, a_j),\ j = 1,2,3:$
capacity$_j$=n$_j$
fixed-charge$_j$=0
flow-cost$_j$=0
$\forall(a_j, s_i),\ j = 1,2,3\quad i\ \forall 1..4:$
capacity$_{j,i}$=∞
fixed-charge$_{j,i}$=0
flow-cost$_{j,i}$=0
$\forall(s_i, \text{sink}),\ i = 1,2,3,4:$
capacity$_i$=n$^{th}{}_i$
fixed-charge$_i$=P$^{idle}{}_i$*e$_i$*PUE$_i$
flow-cost$_i$=c$_i$*p$^{util}{}_i$*e$_i$*PUE$_i$
source
a$_1$
a$_2$
a$_3$
s$_1$
s$_2$
s$_3$
s$_4$
sink
Delays:
d$_{31}$,d$_{41}$ >d^{ref}
d$_{42}$> d^{ref}
d$_{23}$,d$_{13}$> d^{ref}
Front-ends
Data centers

Fig. 5. An example of modeling DAHM for zero migration and switching cost as a FCMCF problem.

[Krumke et al. 1998] but, compared to MIP, more efficient approximation methods have been developed [Carr et al. 2000] (FCMCF can be solved by MIP but not vice versa). We also show that DAHM is NP-hard by reducing an NP-hard subcase of FCMCF to it.

LEMMA 3.1. *The DAHM problem is NP-hard.*

Preliminary. We reduce FCMCF to DAHM. In FCMCF [Krumke et al. 1998] a graph $G = (V, E)$ with nonnegative capacities capacity$_i$ and nonnegative costs w_i for each edge i is given with the edge cost defined on each edge's flow f_i as follows: $w_i =$ flow-cost$_i\, f_i$ + fixed-charge$_i$, when $f_i > 0$ and $w_i = 0$ when $f_i = 0$. The question is whether there is a subset $A \subseteq E$ of the edges of G such that the flow from the source to the sink in (V, A) is at least F and the cost is at most W. FCMCF is known to be NP-hard even on a graph with two nodes and a set of multiple edges between them [Krumke et al. 1998] (this case solves Knapsack).

PROOF. Given a two-node FCMCF instance with a set of multiple edges between them, we construct a DAHM instance as follows: let migration and switching cost be zero: $\beta = 0$ and, $\alpha = 0$ (i.e., the problem becomes memoryless and the index t can be removed), PUE = 1, and electricity cost $e_i = 1$. Also, the delay constraint is relaxed ($d^{\text{ref}} = \infty$). We group the edges such that the capacities capacity$_i$ and costs flow-cost$_i$ and fixed-charge$_i$ are respectively equal within each edge group. Let $|S|$ be equal to the the number of these groups, and let s_i be equal to the number of edges at each group i. Set $c_i = 1$ (i.e., utilization gradient) for all edges. We map the flow F to the number of users, that is, let $|A| = 1$ and $n_1 = F$. Finally, let fixed charge$_i$ and flow-cost$_i$ be p_i^{idle} and p_i^{util} respectively. It is easy to see that the instance of FCMCF has a solution if and only if there is a solution to DAHM by flow split and number of servers of cost at most W. Therefore, DAHM is NP-hard even with zero migration cost, zero switching cost, and no network delay constraint. □

4. SOLUTIONS TO DAHM

4.1. DAHM Solution with Zero Migration and Switching Cost

In this case, a one-epoch DAHM instance can be modeled as an FCMCF. To illustrate the modeling of DAHM as an FCMCF, at epoch t, without loss of generality, we consider a simple case that has $|A| = 3$, and $|S| = 4$. We can make a graph by adding a source and sink node as shown in Figure 5 such that the source is connected to the front-ends by edges whose capacity is equal to the corresponding numbers of the front-ends' users (mapping the service constraint). Each data center is connected to the sink by multiple homogeneous edges such that the number of edges equals the number of available servers (i.e., $s_{i,t}$) (mapping the capacity constraint), where the capacity of each edge equals the maximum affordable workload by each server at data center i (i.e., n_i^{th}) (mapping to the capacity constraint), and the fixed cost and flow-dependent cost are set according to the idle energy cost and utilization energy cost as shown in Figure 5 (in the example $s_{1,t} = 2$, $s_{2,t} = 1$, $s_{3,t} = 1$, and $s_{4,t} = 2$) (mapping the objective function). Finally, the edges between front-ends and data centers are added under no capacity constraint and zero cost. An edge between a data center and a front-end is added if and only if the delay requirement of the front-end can be met by the data center (mapping the delay constraint). A solution flow $F = \sum_{j=1}^{A} n_{j,t}$ to FCMCF can be converted to a DAHM solution by mapping the set of selected edges between data centers and the sink to $y_{i,t}$, and the flow between front-ends and data centers to $x_{i,j,t}$.

The LP relaxation of FCMCF (i.e., converting all integer variables to reals) does not provide a solution that is tight to the optimal [Carr et al. 2000]. However, there has been much work that suggests to use Benders Decomposition [Costa 2005] or branch-and-bound methods to find the exact solution for FCMCF. Both of these solutions have exponential computation cost in the worst case. We used branch-and-bound to find the optimal in our simulation study.

There is a 2-approximation algorithm [Carr et al. 2000], which is based on adding an exponential number of constraints to the problem; consequently, it is very computationally expensive to solve and therefore prohibitive for finding a solution in a time-constrained manner. For that reason, we designed a greedy algorithm, “Greedy”, and show that it has the approximation ratio of $|S|$ under server homogeneity condition (see Appendix A.1). We also show that when, in addition to the preceding condition, the network delay of users is relaxed, the Greedy algorithm becomes a 2-approximation algorithm (see Appendix A.2).

4.1.1. Greedy Algorithm to Solve DAHM for Zero Migration and Switching Cost. The Greedy algorithm associates a Cost Efficiency Metric (CEM) to each data center i at time t as follows: $\text{CEM}_{i,t} = \frac{(p^{idle} + p_i^{util} u_i^{th}) e_{i,t} PUE_i}{n_i^{th}}$, which equals the average cost of a data center normalized to its user capacity. The idea is to use a linear energy cost for data centers and solve the DAHM problem approximately using linear programing and more specifically min-cost flow method (using the energy cost below fixed charge at FCMCF becomes zero, fixed-charge = 0, consequently FCMCF becomes a min-cost flow problem) which has polynomial-time complexity. In this case, the energy cost in Eq. (3) becomes as follows.

$$Cost_t^{energy} = \sum_{i=1}^{|S|} \sum_{j=1}^{|A|} x_{i,j,t} n_{j,t} CEM_{i,t}. \tag{8}$$

Using the aforesaid energy cost, the variable $y_{i,t}$ is removed from the objective function (i.e., Eq. (3)) which will be derived after the solutions for $x_{i,j,t}$s are found as follows: $y_{i,t} = \lceil \frac{\sum_{j=1}^{j=|A|} x_{i,j,t} n_{j,t}}{n_{i,t}^{th}} \rceil$. Similarly the capacity constraint (i.e., Eq. (6)) for each data center and epoch becomes $\frac{\sum_{j=1}^{j=|A|} x_{i,j,t} n_{j,t}}{n_{i,t}^{th}} \leqslant s_{i,t}$.

4.2. DAHM Solution with Nonzero Migration and Switching Costs

Similar to the zero migration cost case, the optimal solution for this case can be obtained using the branch-and-bound technique. However, the optimal solution to the problem can only be obtained offline where all information about the workload and electricity price is available in advance. In this case, DAHM generalizes FCMCF. Similar to the DAHM for zero migration cost case, we can model DAHM at each time as a FCMCF for the nonzero migration cost case. However, the per-epoch solutions of DAHM in the nonzero migration and switching cost case depend on each other (i.e., each epoch's solution depends on the previous epoch's solution), and, to our knowledge, there is no way to connect those FCMCF instances in such a way that migration cost is incurred and flow conservation law is preserved, in order to find a combined optimal solution for the entire period.

We devise the following online algorithms that solve the problem at the beginning of each epoch, t, based on the current hosting state (i.e., $x_{i,j,t-1}$) of the application, the electricity price ($e_{i,t}$) and the next epoch's traffic behavior (i.e., distribution and population of online users).

4.2.1. OnlineMIP Algorithm. The online version of DAHM (Eq. (3)), that is, without summation over time in all the terms in Eq. (3), is solved using the branch-and-bound technique.

The online version of DAHM (Eq. (3)), that is, without summation over time in all the terms in Eq. (3), is solved using branch-and-bound.

4.2.2. Online Greedy Algorithm. The algorithm accounts for the online version of the DAHM problem (Eq. (3)), that is, without summation over time in all the terms in Eq. (3), and uses Eq. (8) as the energy cost model in the objective function. It solves DAHM at each epoch using linear programing.

4.2.3. OnlineCOB, Cost-Oblivious Algorithm. We use a conventional performance oriented load balancing assignment as a baseline algorithm to evaluate the cost efficiency of our approach. In this approach: (i) each area is assigned to a data center whose delay is the least among all other data centers, (ii) load is balanced among data centers whose delay with respect to areas are the same. This is the approach that is currently used for mirror severs [Emens et al. 2003]. Also, the number of servers at each data center is dynamically adjusted at each epoch according to the size of incoming traffic. This algorithm is referred as Online Cost OBlivious, *OnlineCOB* in the rest of the article.

5. SIMULATION STUDY

5.1. Simulation Setup

We simulate a cloud consisting of three data centers. Their characteristics are set according to realistic data. To this end, we assume data centers are located at the following three locations: Atlanta, GA; Houston, TX; and Mountain View, CA, namely DC1, DC2 and DC3, respectively. These locations correspond to the location of three major Google data centers. We used the historical electricity prices for the aforesaid locations [Rao et al. 2010b] (see Figure 1). Note that, in reality, each data center provider may have different electricity price contracts, that is, lower electricity price than households. However, the electricity cost can be defined according to the actual electricity price or the type of energy source (green or brown). The electricity price of Figure 1 is used as an *example* to show the cost saving benefit of DAHM by leveraging electricity cost.

To model the front-end coverage of the data centers we measure the network delay from the simulated data center locations to all U.S. states using `traceroute`. We choose one IP address for each state (e.g., IP address of state universities) and run `traceroute` through three servers of the simulated data center locations (provided by "www.traceroute.org") to all 51 IP addresses. We ran `traceroute` hourly for 24 hours. As we did not find a server in Georgia, the location of DC3, to run `traceroute`, we chose Florida instead. The summary of results, shown in Figure 6, indicates that delay is highly correlated with distance. Also the delay depends on the source network from which `traceroute` is run. Moreover, the daily variation of delays were negligible (within 1 ms).

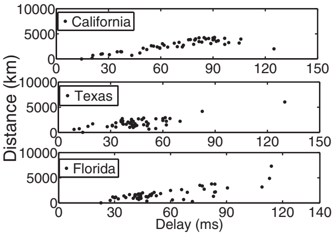

Fig. 6. The network delay between servers from Texas, California, and Florida to all other states in USA versus distance between states.

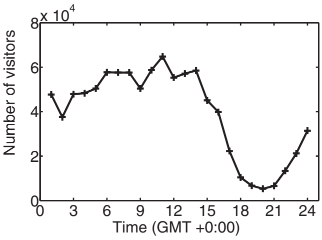

Fig. 7. Hourly number of U.S. online users for an entertainment Web site hosted at GoDaddy.com on 17^{th} March, 2011.

5.1.1. Data Center Types. Three homogeneous (identical) data centers are considered for the simulation with contemporary servers (e.g., IBM Systems x3650 M2: idle power 100 and peak power 320 watt) and very low PUE (we use 1.3, which is the PUE of state-of-the-art data centers [FEMP 2010]). To show the efficiency of the DAHM solution under different energy proportionality of servers, the Idle to Peak power Ratio (IPR) [Varsamopoulos and Gupta 2010] of servers is varied between zero (ideally energy-proportional server) and 0.6 (old servers). The maximum number of servers for each data center is set to 25 which matches the workload intensity range used in the simulations.

To model the utilization of servers, we assume that each online user imposes 0.00005 utilization to each server (i.e., $c = 0.0005$) and that each server can at most handle requests from 2000 online users. The server utilization thresholds, u_i^{th}, are set to 75%.[1] The d^{ref} is set to 66 ms, and data centers' reference delay, d'^{ref} is set to 6 ms [Chen et al. 2005].

5.1.2. Workload Distribution. We used one day (March 17, 2011) of workload trace of an entertainment Web site hosted at *GoDaddy.com*. Using *Google Analytics*, we collected the hourly total number of visitors to the Web site from different USA states (see Figure 7). The workload is scaled up to the data centers' capacity. Also we assume 50% of the users are new users (i.e., $si = 0.5$).

5.1.3. Experiments Performed. We performed different experiments to show the cost saving of DAHM with respect to the energy proportionality of servers (see Section 5.2), migration cost (see Section 5.3), heterogeneity of data centers (see Section 5.4), and workload variation (see Section 5.5). Although DAHM and its online solutions account for the servers' switching cost, we set the switching cost α to zero in all experiments for the sake of simplicity.

[1] This value was determined from anecdotal Web searching. It does not affect the validity of the results but only the amount of savings.

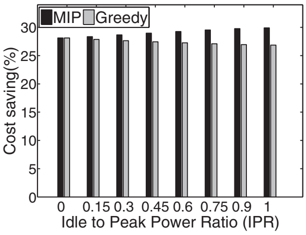

Fig. 8. DAHM cost savings with respect to OnlineCOB over different IPR of servers and zero migration cost (homogeneous DCs).

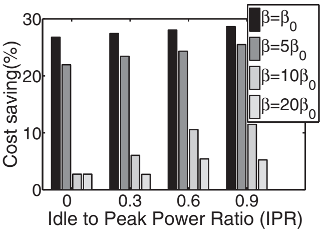

Fig. 9. Cost savings of MIP solution w.r.t. OnlineCOB over different IPR of servers and migration cost β (homogeneous DCs).

We used *GNU Linear Programming Kit (GLPK)* solver under MATLAB 2009, to run the branch-and-bound algorithm on MIP. GLPK is also used to run our Greedy algorithms. All of the cost savings are with respect to OnlineCOB.

Under nonzero migration cost, the offline optimal, namely OfflineMIP, is implemented using branch-and-bound and used to evaluate the proposed online solutions.

5.2. DAHM Optimal Solution versus Greedy under Zero Migration Cost

The DAHM cost saving under different IPR of servers, shown in Figure 8, interestingly indicates that the efficiency of Greedy is better than the theoretical bound (see Appendix A). The same figure shows that at IPR = 0 (energy-proportional case) Greedy and optimal solution incur the same cost. This is expected because in this case DAHM becomes a simple linear programing problem. The DAHM cost saving is due to both leveraging the variation of electricity price and minimizing the number of required servers across all data centers (see Figure 14). The cost saving of the optimal solution increases for higher IPR, because consolidation of servers incurs more cost saving. The results in Figure 8 show the benefit of the DAHM optimal solution over previous workload distribution schemes [Le et al. 2010; Qureshi et al. 2009]; the cost saving of those schemes was maximized under ideally energy-proportional servers and decreased significantly when the servers had an IPR greater than zero.

5.3. DAHM Optimal Solution versus Greedy under Nonzero Migration Cost

To study the migration cost impact on DAHM cost saving, we choose a migration cost comparable to the *reference energy-cost benefit of a migration* denoted by β_0, which is defined as the difference between the energy-cost of the most and the least cost-efficient data centers for one online user. Figure 9 shows that when the migration cost is less than $5\beta_0$, DAHM cost saving only drops from 27% down to 23% (with respect to OnlineCOB). The reason for the small drop is that when the workload share among the data center changes, new users do not have any migration, and that the benefit of migration is usually more than β_0, since it helps to consolidate servers. Therefore, if

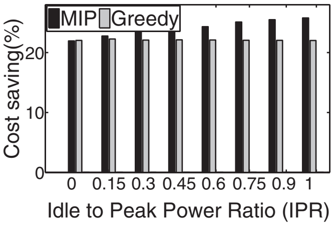

Fig. 10. DAHM cost savings with respect to OnlineCOB under different IPR of servers and nonzero migration cost (homogeneous DCs).

Table II. Data Centers' Characteristics

DC	Elec. price model	Servers' peak power	PUE	case*
DC1	Mountain View, CA.	320	1.3	homogen.
DC1	Mountain View, CA.	400	1.5	heterogen.
DC2	Houston, TX.	320	1.3	homogen. and heterogen.
DC3	Atlanta, GA.	320	1.3	homogen. and heterogen.

*The characteristics of DCs for the homogeneous and heterogeneous case study.

the migration cost is comparable to the cost efficiency difference of data centers, DAHM can still save a significant cost due to reducing the number of servers. The cost saving of DAHM diminishes down to 2.5% for very high migration cost cases: $10\beta_0$ and $20\beta_0$. The reason is that, since the migration is always applied to a portion of users (see Section 3.4), a very high migration cost prevents the workload share changes of data centers for total cost minimization. For the rest of the experiments we adjust $\beta = 5\beta_0$, and refer to it as the nonzero migration cost case.

For nonzero migration cost, neither OnlineMIP nor OnlineGreedy provide an optimal solution. As shown in Figure 10, the cost saving of OnlineMIP is marginally greater than the cost saving of OnlineGreedy. The optimal solution under nonzero migration cost can only be achieved offline. Comparing DAHM offline optimal with respect to the online solutions,[2] we find out that the offline optimal always achieves up to 1% better cost saving over the online solutions with respect OnlineCOB. Its cost saving accumulates with the increase in simulation time. Developing an online algorithm with a competitive bound is left for future work.

5.4. Leveraging Heterogeneity of Data Centers

To investigate the potential saving of heterogeneous data centers and the heterogeneity's effect on the total cost efficiency of DAHM, we make DC1 to be less energy efficient than DC2 and DC3. To this end, the PUE of DC2 is changed to 1.5 and the peak power of servers is changed to 400 W (see Table II). The results in Figures 11 and 12 show that DAHM cost saving increases from the range of 27–30% for the homogeneous data center case to 30–32% for the heterogeneous data center case. Also, in contrast to the case of homogeneous data center (Section 5.3) where Greedy's cost saving for nonzero migration cost decreases with respect to servers' IPR, Greedy's saving in this case increases, yet marginally. The reason is that, in this case, minimizing the number of active servers over the cloud yields more cost saving.

5.5. Leveraging Workload Variation

To investigate how workload variation can be leveraged to save more cost, we used Lagrangian relaxation to move the performance constraint into the objective function

[2] Due to high time complexity of the offline optimal algorithm, we just ran the algorithm for few hours instead of the entire 24 hours.

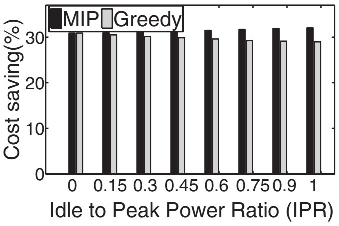

Fig. 11. DAHM cost savings with respect to OnlineCOB under different IPR and zero migration cost (heterogeneous DCs).

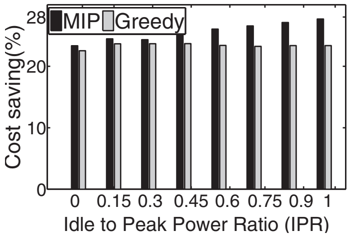

Fig. 12. DAHM cost savings with respect to OnlineCOB under different IPR, and nonzero migration cost (heterogeneous DCs).

Table III. Cost Saving and Delay Trade-Off of DAHM Compared to OnlineCOB

		$\eta = \eta_0$		$\eta = 2\eta_0$		$\eta = 10\eta_0$	
DC case	Algorithm	saving(%)**	viol.(%)**	saving(%)	viol.(%)	saving(%)	viol.(%)
homogen.	MIP	17–25	15–20	12–19	0-6	13–15	0
	OnlineMIP	13–25	0.1–7	11–18	0-2	9–14	0
heterogen.	MIP	22–28	10–18	21–24	1–8	20–22	0
	OnlineMIP	21–26	4–15	19–21	0.5–2.5	15–19	0

*The value of η_0 is set to 0.000001.
**The saving values are given in a range from IPR = 0 to IPR = 1.

(see Eqs. (3) and (7)) and adjust the Lagrangian multiplier (which is η into the number of users whose delay is violated) to force the solution to perform trade-off between energy cost and delay violation minimization. Since the current simulation setup would not yield a lot of delay violations (all DCs have low delays with most states), we change the simulation setup to restrict the data centers' coverage area to at most half of the areas with low delay (this artificially makes areas out of the coverage of a DC to have a delay above the constraint). The latter setup allows investigating the potential cost-performance trade-off under variability of network delays. With this setup, DAHM saves 9–15% cost in the case of no delay violation.

The results in Table III show that allowing delay violations for up to 1% of the users improves the cost saving of DAHM to 13–22% depending on the IPR value and migration cost. This saving can be explained using results in Figures 13 and 14 as follows.

The results in Figure 13 show that workload assignment is tightly correlated with the electricity cost, workload intensity, and the delay constraint. It can be seen in the figure that when the workload intensity of the tested area is low, and the electricity price of the data center where it is assigned increases, DAHM shifts workload to DC2. However, this causes the delay for a few users to be violated.

Figure 14 shows that DAHM violates the delay constraint for when IPR is large because the cost saving of consolidation is higher. Also the figure shows that when

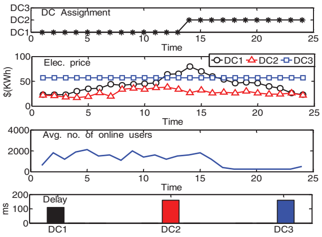

Fig. 13. The data center host and workload density of an area over time (homogeneous DCs).

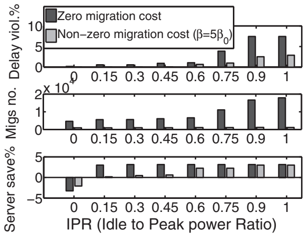

Fig. 14. The performance of OnlineMIP with respect to OnlineCOB under different IPR of servers (homogeneous DCs).

Table IV. Summary of DAHM Problem and Solution Characteristics

mig. cost	problem formulation	optimal sol.	complexity	approx. ratio
zero	FCMCF	Branch&Bound	NP-hard	$\|S\|$ (see A.1), 2 (see [Carr et al. 2000])
nonzero	MIP	Branch&Bound	NP-hard	not known

servers are assumed to be ideally energy-proportional (IPR = 0), DAHM increases the number of active servers compared to OnlineCOB to leverage the electricity price variation. However, under nonzero IPR, DAHM always decreases the total number of servers by 1–4% with respect to OnlineCOB.

5.6. Discussion on DAHM and Its Implementation Issue in Practice

As summarized by Table IV, DAHM is NP-hard. We provide numerical results to evaluate the polynomial-time greedy algorithms and show they can have an approximation ratio equal to the number of data centers for the zero migration cost case. As a practical example, Greedy and GreedyOnline take a fraction of a second to compute the number of servers and workload share for a hypothetical cloud of 100 data centers, whereas MIPOnline takes around half an hour, all running on a 2.8 GHz Intel Pentium system, as performed in a side experiment.

For the homogeneous data center case, DAHM cost saving comes only from leveraging electricity cost and its magnitude depends on the number of available servers in the data centers, the delay constraints, and the algorithm (MIP or Greedy). The maximum cost saving was 40% when there was no limit on the number of servers and delay. We provide numerical results to show how performance of DAHM is affected through the aforementioned parameters in the subsections, that came before. The cost saving difference between Greedy and MIP algorithm diminishes as the workload and number of servers are scaled up, and Appendix A.1 shows the Greedy algorithm approximation ratio for the general case of workload volume.

In practice, different classes of workload may have different SLA and delay requirements. Incorporating the class of workload into the cost model does not change the nature of the problem, yet it needs more parameters to express the problem. Also, it adds the flexibility of DAHM to move workload of a lower class to the most cost-efficient data centers to yield more cost savings. An exhaustive study of this modeling is left for future work.

In our simulation study, we assume that at the beginning of each epoch, the input about workload and electricity price is available; but, in practice, this information should be predicted. Both workload and electricity are predictable, however, the prediction error may marginally decrease the overall cost saving. DAHM can be considered as a central controller and should be frequently updated with information on network delay, electricity price, and history of workload from data centers. Since these data should be sent at each epoch, and each epoch is nominally around half an hour to several hours, its overhead is negligible.

These practical issues will be tested and examined using the BlueTool research infrastructure [Gupta et al. 2011a, 2012], which offers a small data center for experimentation with innovative management schemes such as DAHM.

6. CONCLUSIONS

This article presents problem formulation and algorithms for DAHM, which allow cloud providers to host Web allocation cost efficiently in a dynamic fashion. The problem is formulated according to a cost model that accounts for energy cost of data centers, delay requirement, and traffic behavior of applications as well as live migrations. We show that the problem is generally NP-hard and that in the zero migration cost case, the problem can be modeled as FCMCF. We also show that the polynomial-time Greedy algorithm can provide a performance bound guarantee under some conditions of servers' power consumption (e.g., homogeneous power consumption) (see Table IV). Further, a simulation study is performed using realistic data and we make the following conclusions: (i) dynamic workload and server management minimizes the total number of servers over cloud and yields significant cost savings by removing idle power cost; (ii) dynamic server and workload management can leverage the temporal and spatial variation of electricity price, workload, and data centers' energy efficiency to minimize total cost, and (iii) relaxing the delay requirement of a few users by incorporating the SLA revenue lost in the cost model can increase the total cost efficiency; this is due to: (a) periods of low and high online user population over different areas do not simultaneously happen, and (b) assigning users of areas at periods of low online user population to data centers which are in service for other areas reduces the total number of active servers while it may incur delay violation for a few fractions of the population. Developing online algorithms with competitive ratio with respect to offline optimal is left for our future work.

APPENDIX

A. PERFORMANCE BOUND OF GREEDY ALGORITHM

LEMMA A.1. *The Greedy algorithm (Section 4.1.1) is a $|S|$-approximation for DAHM when these three assumptions hold: (i) zero migration cost, (ii) zero switching cost, and (iii) either (a) servers over the cloud have uniform IPR, or (b) for any two data centers i, k, $CEM_i \leqslant CEM_k \Rightarrow IPR_i \leqslant IPR_k$.*[3]

We use C_i to denote the numerator of CEM (see Section 4.1.1), and overload the notation s_i to denote the data center i. Note that: (i) Greedy prefers data centers with lower CEM because it optimizes the energy cost in Eq. (8), and (ii) we assume that Eq. (8) has a

[3] The 2nd condition is highly likely to happen in practice, since: (i) CEM is dominantly affected by the servers' power model as PUE and electricity price do not vary as much, and (ii) low IPR values also reduce the PUE, and finally (iii) modern servers exhibit low IPR values without any increase in their peak power.

unique optimal solution which is true if no two CEMs are equal (we can add a minute value to one of the equal CEMs to maintain the uniqueness without significantly changing the problem). The latter assumption ensures that as long as an efficient data center has available capacity, no other is used. The proof of the lemma is as follows.

PROOF. Assume all servers within each data center are homogeneous (we argue on the case of heterogeneous data centers right after this proof). Due to the homogeneity of each data center and the linearity of the power model (Section 3.3), it follows that the total power of the assigned servers in a data center is *equivalent* to that of m fully utilized servers (i.e., $u = u^{th}$) and one underutilized server. We denote as $W \subseteq S$ the set of data centers where, for each data center in W, Greedy assigns up to one server which is underutilized, and as $K = S - W$ the set of data centers where Greedy assigns servers of equivalent power of at least one fully utilized server. Assume that Greedy yields $y'_{i,t}$ fully utilized (i.e., $u = u_i^{th}$) and one underutilized server in K. Let the total cost for the fully utilized servers in the set K be k_1, and the total cost for the underutilized servers be k_2, and the total cost of *each* data center in the set W be w, we prove that each of k_1, k_2, and w provide a lower bound on the optimal cost, that is, $Cost(Optimal) \geqslant \max(k_1, k_2, w)$.

Hypothesis 1: *k_1 is a lower bound on the optimal cost*. By contradiction, assume Optimal pays less than k_1, then one of the two must be true.

—*Splitting the workload of one or more fully utilized servers across other data centers achieves cost less than k_1*. Without loss of generality, assume Optimal splits the workload of a fully utilized server at data center 1 (as chosen by Greedy) onto data centers 2 and 3. Since Greedy chose data center 1, it must be true that CEM$_1 \leqslant$ CEM$_2$, and CEM$_1 \leqslant$ CEM$_3$. Assume the q portion of the fully utilized server workload is assigned to a server at data center 2 (and $1-q$ to data center 3). Also consider the most favorable scenario where all servers in all three data centers are truly energy proportional. Then the total cost of the workload becomes

$$\frac{qn_1^{th}C_2}{n_2^{th}} + \frac{(1-q)n_1^{th}C_3}{n_3^{th}} \geq \frac{qn_1^{th}C_1}{n_1^{th}} + \frac{(1-q)n_1^{th}C_1}{n_1^{th}} \geq C_1,$$

where n_i^{th} is the upper number of users *per* server that can be hosted at data center i (see Section 3.3). Hence, the assumed case is contradicted.

—*Merging the workload of one or more fully utilized servers achieves cost less than k_1*. Without loss of generality, assume that Optimal merges the workload of two fully utilized servers in data centers 1 and 2 (as chosen by Greedy) onto data center 3. Also, without loss of generality, assume that CEM$_1 \leqslant$ CEM$_2$ (one data center must be more efficient than the other due to the uniqueness assumption). It follows that $n_3^{th} \geq n_1^{th} + n_2^{th}$. Since Greedy prefers data centers 1 and 2 over data center 3, it also follows that CEM$_1 \leqslant$ CEM$_2 \leqslant$ CEM$_3$. Also, assuming the most favorable scenario where all servers in all three data centers are truly energy-proportional, then we have

$$\frac{n_1^{th}C_3}{n_3^{th}} + \frac{n_2^{th}C_3}{n_3^{th}} \geq \frac{n_1^{th}C_1}{n_1^{th}} + \frac{n_2^{th}C_2}{n_2^{th}} \geq C_1 + C_2,$$

which contradicts the assumed case.

Hence, both the assumptions are contradicted, and Hypothesis 1 holds.

Hypothesis 2: *k_2 is a lower bound on the optimal cost*. Since $k_2 \leqslant k_1$ (for each data center in K, there is up-to-one underutilized server whereas there are one-or-more fully utilized servers), Hypothesis 1 proves this hypothesis as well.

Hypothesis 3: For any *data center in* W, *the cost* w *of its underutilized server* s_{w} *is a lower bound on the optimal cost.* We consider the case that there exist other data centers that respect the delay requirement, otherwise Greedy would have no choice but to match the Optimal and yield the same cost. For the total cost of Optimal to be less than w, either: (i) there is an available server with a lower CEM than the CEM of s_{w}, or (ii) there is a data center on which the optimal solution can put the workload of s_{w} to reduce the cost below w.

Case (ii) contradicts with the definition of Greedy: this would be possible if the server had a lower IPR than s_{w}, which contradicts with Lemma A.1's condition-iii-a or iii-b.

For case (i), we observe that, for Greedy to select s_{w} instead of any other eligible server with lower CEM, it must be so because all other eligible servers are fully utilized. Assume, by contradiction, that the optimal cost is less than w. By Hypothesis 1, we know that the optimal solution can not pay less than Greedy for the fully utilized servers (i.e., k_1). Therefore, for Optimal to achieve a lower total cost, it must merge the workload of some of fully utilized servers and the underutilized one (i.e., server s_{w}) onto any other server. This is possible only if the IPR of the target server is less than of that s_{w}, which contradicts with Lemma A.1's condition iii-a or iii-b.

Combining the preceding hypotheses, $Cost(Optimal) > \max(k_1, k_2, w)$. The lower bound on total cost for the data centers in the set W is $|W|Cost(Optimal)$. We know that $Cost(Greedy) = k_1 + k_2 + |W|w$, then according to Hypotheses 1, 2 and 3, it follows that

$$\frac{Cost(Greedy)}{Cost(Optimal)} = \frac{k_1+k_2+|W|w}{Cost(Optimal)}$$
$$\leqslant \frac{Cost(Optimal)+Cost(Optimal)+|W|Cost(Optimal)}{Cost(Optimal)} \overset{W\subseteq S}{\leqslant} \frac{(|S|+1)Cost(Optimal)}{Cost(Optimal)} = |S|+1. \quad (9)$$

□

For the *heterogeneous case* of data centers, it is easy to see that the approximation ratio is the number of the classes of servers in the cloud. A tight example is as follows: assume $|A| = 3$ and $|S| = 3$, where each data center has only one available server and that all have IPR = 1 (no utilization-dependent cost). Further, assume s_1 and s_2 have the same CEM, where the total cost and capacity of a fully utilized server for them is 1 and 10, whereas they are $1 + \varepsilon$, and 10 respectively for s_3. (i.e., $\mathrm{CEM}_1 = \mathrm{CEM}_2 = \frac{1}{10} < CEM_3 = \frac{1+\varepsilon}{10}$). Also assume s_1 can only respect the delay requirement of a_1 workload, similarly s_2 can only respect the delay requirement of a_2. But, the delay requirement of all areas can be respected by s_3. Finally, assume each area has only one user. Greedy selects s_1 to provide service for a_1 workload, s_2 for a_2, and s_3 for a_3, hence it incurs $3 + \varepsilon$ cost in total. However, the optimal solution selects only s_3 to provide service for all areas, and incurs only $1 + \varepsilon$ cost. Note that the worst-case situation happens only if utilization-dependent cost of servers is zero, which is not the case in practice.

Proposition A.2. *Greedy, under the conditions of Lemma A.1, is a 2-approximation ratio when there is no network constraint delay or it is universally satisfied, that is,* $d_{ij} = d_i'^{ref} + d''_{i,j} \leqslant d^{ref}$, $\forall\, i = 1 \ldots |S|$ *and* $j = 1 \ldots |A|$.

Proof. In this case, Greedy incurs at most one underutilized server. By contradiction, assume that there are two or more underutilized servers in S. If the data centers containing the underutilized servers are of equal CEM, then we can merge their workload and result in only one underutilized server without altering the total cost, thusly contradicting the assumption. Conversely, if those data centers have unequal CEMs, it will contradict with the definition of Greedy which does not assign another data center before it fully uses the one with smaller CEM. Hence, Greedy yields only one underutilized server; specifically, either $|W| = 1$ and $k_2 = 0$, or $|W| = 0$ and $k_2 \neq 0$, or $|W| = 0$

and $k_2 = 0$. Therefore, we can safely conclude from Lemma A.1 that the approximation ratio of Greedy in this case is 2. □

ACKNOWLEDGMENTS

The authors would like to thank the special issue's guest editors, Partha Pande and Amlan Ganguly, and the anonymous reviewers for their insightful suggestions.

REFERENCES

ABBASI, Z., VARSAMOPOULOS, G., AND GUPTA, S. K. S. 2012. TACOMA: Server and workload management in Internet data centers considering cooling-computing power trade-off and energy proportionality. *ACM Trans. Archit. Code Optim. 9*, 2, Article 11.

ABBASI, Z., VARSAMOPOULOS, G., AND GUPTA, S. K. 2010. Thermal aware server provisioning and workload distribution for internet data centers. In *Proceedings of the ACM International Symposium on High Performance Distributed Computing (HPDC'10).* 130–141.

BARROSO, L. A. AND HÖLZLE, U. 2007. The case for energy-proportional computing. *Comput. 40*, 33–37.

BESKOW, P., VIK, K., HALVORSEN, P., AND GRIWODZ, C. 2009. The partial migration of game state and dynamic server selection to reduce latency. *Multimed. Tools Appl. 45*, 1, 83–107.

BUCHBINDER, N., JAIN, N., AND MENACHE, I. 2011. Online job-migration for reducing the electricity bill in the cloud. *Netw. 6640*, 172–185.

CARR, R., FLEISCHER, L., LEUNG, V., AND PHILLIPS, C. 2000. Strengthening integrality gaps for capacitated nework design and covering problems. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics*, 106–115.

CHASE, J., ANDERSON, D., THAKAR, P., VAHDAT, A., AND DOYLE, R. 2001. Managing energy and server resources in hosting centers. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP'01).* 103–116.

CHEN, Y., DAS, A., QIN, W., SIVASUBRAMANIAM, A., WANG, Q., AND GAUTAM, N. 2005. Managing server energy and operational costs in hosting centers. *SIGMETRICS Perform. Eval. Rev. 33*, 1, 303–314.

CHUN, B. G., IANNACCONE, G., IANNACCONE, G., KATZ, R., LEE, G., AND NICCOLINI, L. 2010. An energy case for hybrid datacenters. *ACM SIGOPS Oper. Syst. Rev. 44*, 1, 76–80.

COSTA, A. 2005. A survey on benders decomposition applied to fixed-charge network design problems. *Comput. Oper. Res. 32*, 6, 1429–1450.

EMENS, M., FORD, D., KRAFT, R., AND TEWARI, G. 2003. Method of automatically selecting a mirror server for web-based client-host interaction. Patent 6,606,643.

FEMP AND GSA. 2010. Quick start guide to increase data center energy efficiency. Tech. rep., General Services Administration (GSA) and the Federal Energy Management Program (FEMP).

GUPTA, S. K. S., GILBERT, R. R., BANERJEE, A., ABBASI, Z., MUKHERJEE, T., AND VARSAMOPOULOS, G. 2011a. GDCSim: A tool for analyzing green data center design and resource management techniques. In *Proceedings of the International Green Computing Conference (IGCC'11).* IEEE Press.

GUPTA, S. K. S., MUKHERJEE, T., VARSAMOPOULOS, G., AND BANERJEE, A. 2011b. Research directions in energy-sustainable cyber-physical systems. *Elsevier Sustain. Comput. 1,* 1, 57–74.

GUPTA, S. K. S., VARSAMOPOULOS, G., HAYWOOD, A., PHELAN, P., AND MUKHERJEE, T. 2012. *Handbook of Energy-Aware and Green Computing*. No. 45, Chapman and Hall/CRC, Chapter BlueTool: Using a computing systems research infrastructure tool to design and test green and sustainable data centers.

HSU, C. AND POOLE, S. 2011. Power signature analysis of the SPECpower_ssj2008 benchmark. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS'11).* IEEE, 227–236.

KOOMEY, J. G., BELADY, C., PATTERSON, M., SANTOS, A., AND LANGE, K.-D. 2009. Assessing trends over time in performance, costs and energy use for servers. Tech. rep., Microsoft Corp. and Intel Corp.

KRIOUKOV, A., MOHAN, P., ALSPAUGH, S., KEYS, L., CULLER, D., AND KATZ, R. 2010. NapSAC: Design and implementation of a power-proportional web cluster. In *Proceedings of the 1st SIGCOMM Workshop on Green Networking.* ACM, New York, 15–22.

KRUMKE, S., NOLTEMEIER, H., SCHWARZ, S., WIRTH, H.-C., AND RAVI, R. 1998. Flow improvement and network flows with fixed costs. *OR 98.*

KUMAR, K. AND LU, Y.-H. 2010. Cloud computing for mobile users: Can offloading computation save energy? *Comput. 99,* 51–56.

KUSIC, D., KEPHART, J. O., HANSON, J. E., KANDASAMY, N., AND JIANG, G. 2009. Power and performance management of virtualized computing environments via lookahead control. *Cluster Comput. 12*, 1–15.

LE, K., BILGIR, O., BIANCHINI, R., MARTONOSI, M., AND NGUYEN, T. 2010. Managing the cost, energy consumption, and carbon footprint of internet services. *SIGMETRICS Perform. Eval. Rev. 38*, 1, 357–358.

LIN, M., WIERMAN, A., ANDREW, L., AND THERESKA, E. 2011. Dynamic right-sizing for power-proportional data centers. In *Proceedings of the IEEE InfoCom Conference on Computer Communications, Joint Conference of the Computer and Communications Societies (InfoCom'11).* 10–15.

LIU, Z., LIN, M., WIERMAN, A., LOW, S.-H., AND ANDREW, L. L. H. 2011. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS Conference on Measurement and Modeling of Computers*. ACM Press, New York.

QURESHI, A., WEBER, R., BALAKRISHNAN, H., GUTTAG, J., AND MAGGS, B. 2009. Cutting the electric bill for internet-scale systems. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications.* 123–134.

RAO, L., LIU, X., ILIC, M., AND LIU, J. 2010a. MEC-IDC: Joint load balancing and power control for distributed internet data centers. In *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems.* 188–197.

RAO, L., LIU, X., XIE. L., AND LIU, W. 2010b. Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *Proceedings of the IEEE InfoCom Conference on Computer Communications, Joint Conference of the Computer and Communications Societies (InfoCom'10).* 1–9.

VARSAMOPOULOS, G., ABBASI, Z., AND GUPTA, S. K. S. 2010. Trends and effects of energy proportionality on server provisioning in data centers. In *Proceedings of the International Conference on High Performance Computing (HiPC'10).* 1–11.

VARSAMOPOULOS, G. AND GUPTA, S. K. S. 2010. Energy proportionality and the future: Metrics and directions. In *Proceedings of the IEEE International Conference on Parallel Processing Workshops (ICPPW).*

Received July 2011; revised November 2011; accepted January 2012