Dynamic Virtual Machine Allocation in Cloud Server Facility Systems with Renewable Energy Sources

Dimitris Hatzopoulos
University of Thessaly,
Greece

Iordanis Koutsopoulos
Athens University of Economics and
Business and CERTH,Greece

George Koutitas
University of Thessaly,
Greece

Ward Van Heddeghem
Ghent University - iMinds,
Belgium

Abstract—**This paper explores the problem of virtual machine (VM) allocation in a network of cloud server facilities which are deployed in different geographical areas. Each cloud server facility is connected to the conventional power grid network and in addition it is supported by an attached renewable energy source (RES). We address the problem of energy-efficient task allocation in the system in the presence of a time-varying grid energy price and the unpredictability and time variation of provisioned power by the RES. The objective is to reduce the total cost of power consumption for the operator. The key idea is to match the VM load with the RES provisioned power. Each request for a task to be executed in the cloud is associated with a VM request with certain resource requirements and a deadline by which it needs to be completed. The cloud provider has to create a VM with the resource requirements of the request and to execute the VM before the deadline. We propose an online algorithm with given look-ahead horizon, in which the grid power prices and patterns of output power of the RESs are known a priori and we compare it with a greedy online algorithm. Numerical results on real traces of cloud traffic and renewable source generation patterns are encouraging in terms of the performance of our techniques and motivate further research on the topic.**

Index Terms - Cloud Systems, Renewable Energy Sources, Dynamic Task Allocation

I. INTRODUCTION

Cloud computing is a technology employed to provide services over a physical network infrastructure to remote customers. Cloud computing systems facilitate flexible task execution by allocating resources (storage, CPU, etc) on demand. The cloud provider is responsible for managing the infrastructure through efficient resource allocation and task request scheduling for virtual machines. A virtual machine is a tightly isolated software container that can run its own tasks as if it were a physical machine. A virtual machine behaves exactly like a physical machine and contains its own virtual (i.e software-based) CPU, RAM, hard disk and network interface card.

Depending on the knowledge of the cloud provider about the parameters and the stochasticity of the system, scheduling and allocation algorithms serve the service requests so as to optimize an objective like the operational cost or the power consumption. Nowadays, a big portion of the energy consumed

by end-users is shifted to the core network of the cloud server infrastructure. This fact motivates our work in greening the cloud infrastructure by using RESs attached to the server facilities.

We study the problem of optimal VM allocation in a set of cloud server facilities. Our objective is to minimize the total cost paid by the cloud provider to the main grid in order to support the system with the necessary power to carry out VM execution. The set of VMs under execution at a particular server form the server load and affect the power consumption of the components of that server, and primarily the portion of power consumed for the CPU utilization. Our system consists of a cloud provider that owns a geographically distributed cloud server facilities and exposes this infrastructure to clients which do not have the resources to execute their tasks. Whenever a client asks for resources, the cloud provider creates a VM and selects one of his cloud server facilities to host it. All the requests initially arrive at a central dispatcher (e.g., a web server), and each request is determined by a number of floating-point operations (FLOPs) that have to be executed before a deadline.

Cloud server facilities may require large amounts of energy to operate, depending on the prevailing environmental conditions and the dynamic load of tasks under execution. As a general observation [1], a significant portion of the power consumption of legacy data centers is associated with the cooling needs and losses of the power provisioning equipment. The rest of the percentage above is modeled by an approximately linear function of the served traffic [2].

To meet the energy demands of the cloud server facilities, an environmental-friendly solution is the deployment of RESs. The RESs reduce dependence on the main power grid and hence, if they are appropriately exploited, they can lead to a significant cost reduction. The main advantages of RESs are that they incorporate an initial capital expenditure for the purchase and deployment but they have very low operational expenses (mainly maintenance), and thus in the long-run their provisioned energy cost is very low. However, RESs have unpredictable behavior and provide a time-varying output power which, in some cases, is insufficient to support the operation of the cloud server facility, if the load in the latter is high enough. In this case, the required power will have to be
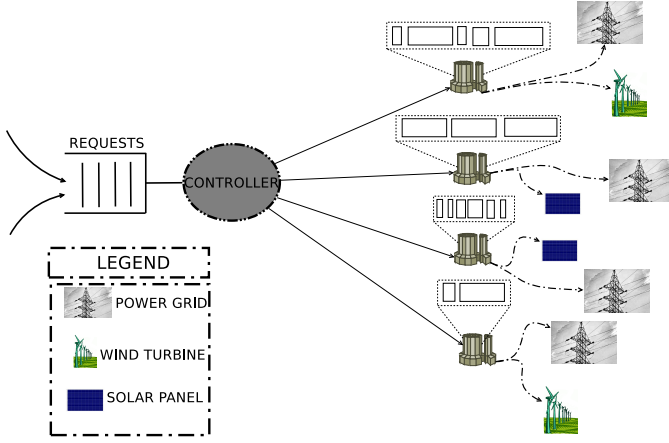
Fig. 1. Requests arrive at a central controller which allocates them to cloud server facilities, each of which fulfills the energy demands from a RES and the power grid.

TABLE I
NOTATION TABLE

Symbol	Meaning
$\mathcal{D}$	The set of the cloud server facilities
B_i	The total processing capacity of cloud server facility i
F_j	The amount of FLOPs in the VM request j
d_j	The deadline imposed in the VM request j
c_j	The processing capacity of VM request j
α_j	The arrival time of VM request j
τ_j	The running time of VM request j
$R_i(t)$	The power generation of the RES plugged in cloud server facility i
$p_i(t)$	The price of the power from the power grid in cloud server facility i
$L_i(t)$	The load of the cloud server facility i

provided by the main power grid in a price that also fluctuates with time.

The rest of the paper is structured as follows: in section II, we present the model and assumptions in more detail and in section III we formulate the problem and describe the system controls. In section IV we propose the VM allocation algorithms and in section V we discuss their performance.

II. SYSTEM MODEL

We consider a set $\mathcal{D}$ of D geographically distributed cloud server facilities $\mathcal{D} = \{1, \ldots, D\}$ and a set $\mathcal{B}$ of B resources, $\mathcal{B} = \{1, \ldots, B\}$. The set of the resources may include processing capacity, storage, memory, bandwidth etc. For example, a request for a database service like the Amazon relational database service [3], needs processing capacity (expressed as Elastic Compute Units), storage (GByte), memory (GByte) and I/O capacity (request rate).

In this work, we concentrate on one type of resource, the processing capacity, for the reason that the CPU is the major component that consumes power, but our approach can be extended to include more types of resources. Assume that B_i is the processing capacity of cloud server facility i, in **fl**oating-point **o**perations **p**er **s**econd (FLOPs/sec).
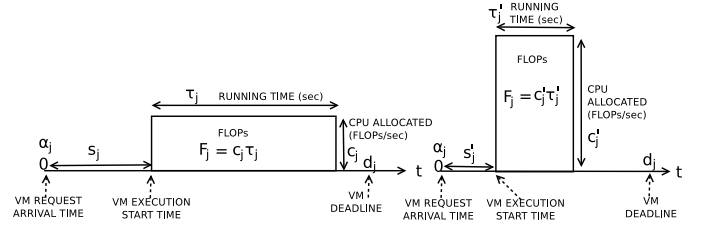

Fig. 2. Two feasible realizations of processing capacity allocation, running time and start execution time of a VM request with $\alpha_j = 0$.

A. Service Requests

Each VM request j arrives at the central queue at time α_j and is specified by a number of (FLOPs) F_j and a deadline d_j by which the execution of the VM should be finished. The cloud provider is responsible to create a VM for that request and allocate it to a cloud server. The created VM will have processing capacity c_j (FLOPs/sec) and will be hosted for a time τ_j (sec). The cloud provider can either allocate a VM with low processing capacity yet enough to finish the task before the deadline, or it can allocate a VM with a high processing capacity to finish the task as soon as possible. This flexibility is depicted in figure 2.

B. Renewable Source Generation and Power Consumption at the Servers

Each cloud server facility $i \in \mathcal{D}$ has a RES installed that produces a time-varying amount of power at time t, denoted by $R_i(t)$. The amount $R_i(t)$ can be known only for a short time period. This knowledge is usually extracted from weather-forecast data. Each cloud server facility i can draw its energy either from the RES or from the main power grid. For the latter we assume a time-varying price per unit of power at each time t, denoted by $p_i(t)$. Also, each cloud server facility i has a load $L_i(t)$ at time t and in order to support that load, it needs power $f(L_i(t))$ where $f(\cdot)$ is an one-to-one function that maps the load to power consumption.

Current cloud server facilities consist of power-saving servers that incorporate frequency and voltage reduction in order to decrease the power consumption [4]. These power-saving features adjust the power consumption of an idle server to its actual load and characterize the relationship of the power consumption and the load of a server, usually as a quadratic function $f(L) = P_{idle} + \alpha L^2, \alpha > 0$ (Fig. 2 of [4]). Table I provides a summary of the notation used in the paper.

III. PROBLEM FORMULATION AND SYSTEM CONTROLS

We are given a set of requests $\mathcal{Y} = \{1, \ldots, N\}$ in a time horizon H and for each request $j \in \mathcal{Y}$, the arrival time α_j, the processing requirements F_j and the deadline d_j. We want to find the start execution time s_j, the running time τ_j and the processing capacity c_j for each created VM. The set of all feasible solutions is:

$$\mathcal{F}_j = \{(s_j, \tau_j, c_j) : \alpha_j \leq s_j \leq d_j - \tau_j \text{ and } c_j \tau_j = F_j\}. \quad (1)$$

It is worth mentioning that once a VM starts its execution in a cloud server facility, it does not migrate to another cloud server facility. Furthermore, we assume the allocated capacity c_j remains fixed for the entire duration of the VM execution.

A. One Server Facility

In the case of a single server facility, the price of a unit of power from the main grid at time t is $p(t)$ and the RES power generation is $R(t)$. Also the load is described by:

$$L(t) = \sum_{j:s_j \leq t \leq s_j + \tau_j} c_j, \quad (2)$$

and the total power consumption by $f(L(t))$. The optimal solution for every request in $\mathcal{Y}$ is given by:

$$\min_{(s_j, \tau_j, c_j) \in \mathcal{F}_j} \int_0^H p(t) \left[f(L(t)) - R(t) \right]^+ dt, \quad (3)$$

where q^+ equals q if $q > 0$ otherwise it is 0.

B. Multiple Server Facilities

In the case of more than one server facility we have to introduce an assignment parameter x_{ij} indicating the selection of the server facility i to execute the VM j, and let $\mathbf{x} = (x_{ij} : i = 1, \ldots, D, j = 1, \ldots, N)$. Each VM can be assigned to one cloud server facility, and the assignment variable is described as:

$$x_{ij} = \begin{cases} 1, & \text{if VM } j \text{ assigned to cloud server facility } i \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The load of every cloud facility i is described by:

$$L_i(t) = \sum_{j:s_j \leq t \leq s_j + \tau_j \text{ and } x_{ij}=1} x_{ij} c_j. \quad (5)$$

The optimal solution for every request in $\mathcal{Y}$ is given by:

$$\min_{(s_j, \tau_j, c_j) \in \mathcal{F}_j, \mathbf{x}} \sum_{i=1}^D \int_0^H p_i(t) \left[f(L_i(t)) - R_i(t) \right]^+ dt \quad (6)$$

$$\text{subject to:} \quad \sum_{i=i}^D x_{ij} = 1 \ , j = 1, \ldots, N. \quad (7)$$

IV. VIRTUAL MACHINE ALLOCATION ALGORITHMS

In this section, we propose an online algorithm that operates upon arrival of a VM request and determines the cloud server facility to which a VM it will be allocated. We consider two possibilities for the knowledge about the RESs power generation and the price of grid power.

A. Online Algorithm with Look-ahead Window T

We propose an algorithm that allocates a VM to a cloud server facility, and it decides on the VM starting time, execution time and processing capacity of the VM. The RES power supply pattern and the power grid prices are assumed to be known for a look-ahead horizon $T \leq H$. This assumption is valid because the RES power supply may often be predictable (e.g. the power of a photovoltaic panel) for a look-ahead time horizon of some minutes or even hours. Also, the price of the

power from the power grid does not change fast and thus the assumption of the apriori knowledge for the next few hours is valid. The additional power needed to host VM j in a cloud server facility i, assuming there exist enough resources available, at time t is:

$$C_{j \to i}(t, c_j) = \left[f(L_i(t) + c_j) - f(L_i(t)) - R_i(t) \right]^+. \quad (8)$$

The additional power is zero if there exists enough amount of renewable power $R_i(t)$. Here we implicitly assume that energy storage does not take place. The additional cost of hosting VM j in a cloud server facility i starting at time s_j while taking into account the price at i and the running time of the VM, is:

$$c_{j \to i}(s_j, \tau_j, c_j) = \int_{s_j}^{s_j + \tau_j} p_i(t) C_{j \to i}(t, \frac{F_j}{\tau_j}) dt. \quad (9)$$

Given that the new VM j will be assigned to cloud server facility i, the optimal time to start s_j^*, the optimal running time τ_j^* and the optimal processing capacity c_j^* are given from:

$$\min_{(s_j, \tau_j, c_j) \in \mathcal{F}_j} c_{j \to i}(s_j, \tau_j, c_j). \quad (10)$$

For $t \in [T, d_j]$ we use the last known values of $R_i(T)$ and $p_i(T)$. We compute one cost with (10) for every cloud facility. Out of the D cloud server facilities, we choose the one for which the cost is minimum.

B. Online Greedy Algorithm (T = 0)

We now assume only instantaneous knowledge (at time t) of the RES generation pattern and price per unit of power from the main grid. An online greedy algorithm creates a VM to the cloud server facility that offers the cheapest additional hosting cost by taking into account only these instantaneous values. The new VM starts its execution immediately (i.e. at $t = \alpha_j$) and finishes on the deadline d_j, and the processing capacity is determined accordingly as $F_j/(d_j - \alpha_j)$. Then the cloud server facility i^* that will host the VM determined at the request is:

$$i^* = \arg \min_{i \in \{1, \ldots, D\}} \left[p_i(\alpha_j) C_{j \to i}(\alpha_j, \frac{F_j}{d_j - \alpha_j}) \right]. \quad (11)$$

V. PERFORMANCE EVALUATION

In this section we present the performance of our allocation algorithms. We implement a Java simulator.

A. Randomized allocation

In order to assess the performance of our algorithms, we compare it with a benchmark random algorithm which assigns each incoming request to a random cloud server facility that has the resources to host it.
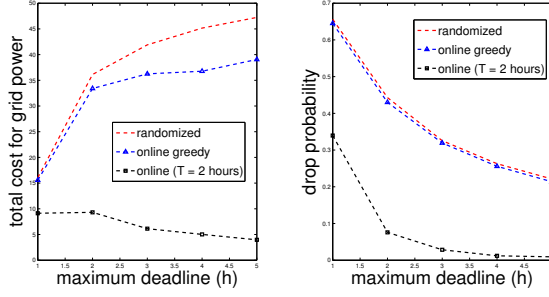
Fig. 3. Total cost and request drop probability vs maximum deadline of requests.

Fig. 4. Total cost vs arrival rate multiplication factor.

B. Input Data

In order to test our algorithm we use real data traces. In case of wind-turbine generation, we use data from [5] and for solar panel generation we use a Gaussian-shaped function with peak at mid-day. The peak of the Gaussian-shaped function of every solar RES is equal to 1.2 times the power the cloud server consumes when it is fully loaded. We assume that the arrival rate for VM executing dynamically changes per hour and we use as arrival rates the following normalized values from [6], namely $\lambda = [0.6, 0.52, 0.4, 0.28, 0.2, 0.17, 0.16, 0.175, 0.23, 0.32, 0.45, 0.6, 0.64, 0.69, 0.72, 0.75, 0.78, 0.8, 0.83, 0.9, 0.97, 0.94, 0.82\ 0.71]$, where every value is the average arrival rate during one hour. The first value corresponds to 21.00. The number of the FLOPs and the deadline of every arriving request follow a uniform distribution whose limits are specified in every experiment.

C. One-day study

In the one-day study, the price of electricity from the power grid is fixed for each cloud facility and equal to 0.1265 euros per kWh. We create $D = 5$ geographically distributed cloud facilities with a capacity of $1000,1500,800,1200,500$ processing units (FLOPs/sec) respectively, and we assume that 3 of them have solar panel RES and 2 of them have wind-turbines. Each facility is located in a different time zone (UTC 0,UTC -7,UTC 3 ,UTC 5,UTC 12), and that means that the solar panels are not at their peak generation at the same time.

Figure 3 shows how the deadlines of requests affect the operational cost of the cloud server facility and the drop probability. The drop probability defines the case where the requested resources to one of the cloud servers can not be served. The randomized and the online greedy algorithms do not have the ability to postpone VM hosting for the near future, and that decreases chances of hosting the VM if the deadline is too near. On the other hand, the online algorithm with look-ahead window utilizes system resources more effectively because it has flexibility in terms of the starting time the VM. This reduces the probability of dropping the request.

In the same setting of 5 servers and one-day horizon we assume that each arriving request imposes a deadline with limits [1,4] hours and number of FLOPs with limits [100,2000].
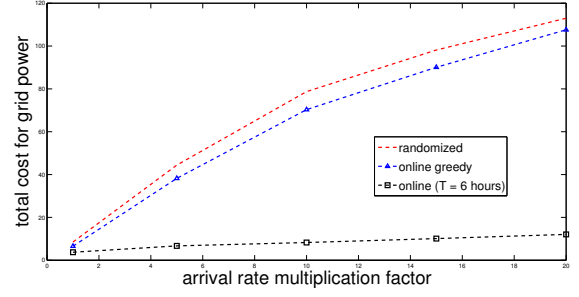
Figure 4 presents the average total cost of the cloud facility as a function of the arrival rate, over 50 independent simulation runs. The x-axis is the multiplier by which we multiply vector λ. The online algorithm with look-ahead window of 6 hours leads to reduced cost because it exploits the RESs in a better way since it controls both the start of execution of each task and the allocated processing capacity.
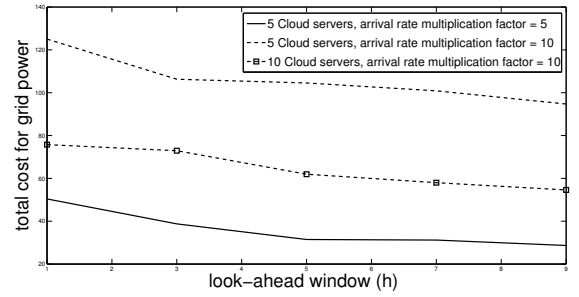

Fig. 5. One day horizon. Cloud servers have equal capacity 5000 FLOPs/second, each arriving request has a deadline in [1, 20] hours.

Figure 5 depicts the reduction of power cost as the length of the look-ahead window increases. Furthermore because of the geographical dispersion of the cloud facilities the power cost increases when both the number of cloud facilities and the arrival rate multiplication factor double.

D. Follow-the-sun study

The setting of that experiment consists of 4 cloud servers with equal capacity of 5000 FLOPs/second which are located in different time-zones and are supported by solar panels. The arrival rates are multiples of those in the vector λ by a factor of 15. For each request, the deadline is specified by a uniform distribution in [1,20] hours and the number of FLOPs in [100,1000]. Figure 6 shows how the online algorithm with a certain look-ahead window (2 and 6 hours) selects the cloud facility which offers the minimum cost. Also, figure 6, depicts the fact that the bigger the look-ahead window is, the more efficient the VM allocation is.

VI. RELATED WORK

There has been a lot of work on VM allocation in cloud computing environments. However, to the best of our knowl-
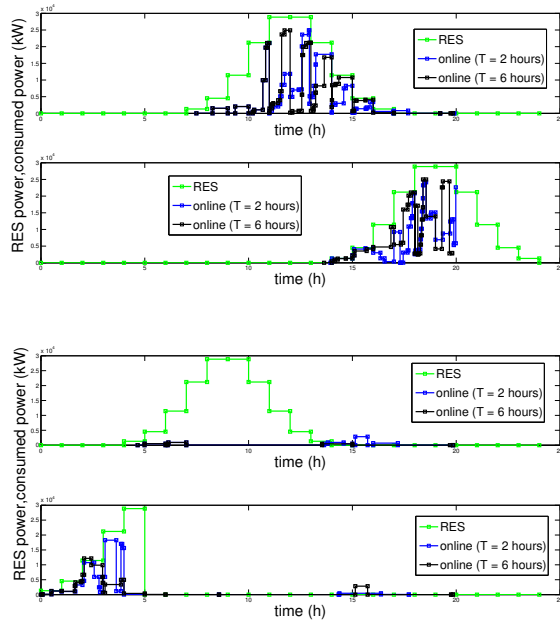
Fig. 6. Power consumption and RES power supply (kW) during one day for 4 cloud facilities.

edge, our work is the first to deal with the problem of VM allocation under a deadline constraint in a cloud environment with unpredictable power supply from RESs so as to minimize the total cost of the power supply from the power grid.

The authors of [7] propose one preemptive and one non-preemptive throughput-optimal scheduling algorithm. There are specific VM configurations. Requests for these configurations arrive and are queued at a central scheduler. The scheduler dispatches a request to a server when the server has enough resources to host the requested VM. The work of [8] proposed a two time-scale algorithm for power cost reduction in geographically distributed data centers. The control variables are the VM assignment, the number of the servers that would be turned on in order to process the queued jobs and the processing rate for all active servers in the cluster.

A different line of works is [9], which focuses on reducing the electricity bill of commercial web search engines operating on data centers that are geographically far apart. To achieve that, the authors propose a technique based on the observation that energy prices and query workloads show high spatiotemporal variation. The authors of [10] characterize the variation due to fluctuating electricity prices and argue that existing distributed systems should exploit this variation to get economic gains. In the area of RESs, the authors of [11] propose the adoption of accurate solar and wind energy predictors more efficient than state-of-the-art time series models.

In addition, the authors of [12] and [13] evaluate the impact of geographical load balancing, the optimal mix of RESs and the role of storage in order to investigate the feasibility of powering internet-scale systems using entirely renewable energy. The authors of [14] model the energy flows in a data

center and optimize its holistic operation. They design a workload manager that schedules workload and allocates resources within a data center according to time-varying power supply and cooling efficiency using RESs and demand predictions.

VII. CONCLUSION

We studied the problem of VM allocation and scheduling in a set of cloud facilities. Each facility was characterized by a time-varying RES and a fluctuating price per unit of power obtained from the main grid. They key idea of our approach for the VM allocation and scheduling (VM start time and processing capacity allocation) was to load each server in such a way that the resulting power consumption matches the RES generation pattern. If the server load exceeded the RES power, then the minimum price per unit of power drawn from the main grid led our decision. In this work, the model for the VM was a simple one; in the future we plan to focus on defining a more detailed model that captures possibly multitenancy and other realistic phenomena encounted in the cloud.

VIII. ACKNOWLEDGMENTS

REFERENCES

[1] G. Koutitas and P. Demestichas, "Challenges for energy efficiency in local and regional datacenters," in *Journal Green Engineering, vol. 1, no.1,*, 2010.
[2] L. Barroso and U. Hlzle, *The data center as a computer: An introduction to the design of warehouse-scale machines*. Morgan and Claypool, 2009.
[3] "Amazon relational database service." http://aws.amazon.com/rds/.
[4] M. Blackburn, *Five ways to reduce data center server power consumption*. The Green Grid, White Paper, 2008.
[5] "Centre for renewable energy sources and saving." http://www.cres.gr/kape/index_gr.htm.
[6] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of datacenter traffic: Measurements and analysis," in *Proc. ACM IMC*, 2009.
[7] S. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proc. IEEE INFOCOM*, 2012.
[8] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. J. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. IEEE INFOCOM*, 2012.
[9] E. Kayaaslan, B. Cambazoglu, R. Blanco, F. Junqueira, and C. Aykanat, "Energy price-driven query processing in multi-center web search engines," in *Proc. ACM SIGIR*, 2011.
[10] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *Proc. ACM SIGCOMM*, 2009.
[11] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing, "Utilizing green energy prediction to schedule mixed batch and service jobs in data centers," in *Proc. ACM HOTPOWER*, 2011.
[12] Z. Liu, M. Lin, A. Wierman, S. Low, and L. Andrew, "Geographical load balancing with renewables," in *Proc. ACM SIGMETRICS*, 2011.
[13] Z. Liu, M. Lin, A. Wierman, S. Low, and L. H. Andrew, "Greening geographical load balancing," in *Proc. ACM SIGMETRICS*, 2011.
[14] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *Proc. ACM SIGMETRICS*, 2012.