A Transactional Framework for Broadening Access to Geo-Diversification

Jared Polonitza
Mathematics and Computer Science
University of Puget Sound
Tacoma, WA

David Chiu
Mathematics and Computer Science
University of Puget Sound
Tacoma, WA

Bin Ren
Computer Science
College of William and Mary
Williamsburg, VA

***Abstract*—This paper establishes a transactional energy market in which *any* data center can participate. Like a stock-market exchange, we propose a framework in which data centers can trade energy usage (in the form of jobs) for monetary value. The proposed framework allows each data center to monitor multiple parameters, including the current energy prices, budgets, and job execution states. These parameters inform the construction of models to help participating data centers optimize various cost, profit, and job-performance objectives to manage the risks of market participation. In our feasibility study, market participants mutually benefit by increasing general revenue through either a reduction of energy costs and/or through the successful completion of more jobs. Using real energy-pricing data, in our simulated experiment of 100 participating data centers, we observe a significant cost reduction leading to an average increase of 17.8% profit margins.**

I. Introduction

To minimize total operating costs, major data-center operations can exploit the variability of energy prices, *i.e.*, local marginal price (LMP). A more ambitious approach to exploit price variability is through *geo-diversification*, in which an organization builds multiple data center locations across disparate geographical regions. When energy is cheap at one location, computational workloads can be scheduled more aggressively and even migrated there from locations experiencing higher prices.

However, due to the high cost required to geo-diversify, this practice is inaccessible to the vast majority of existing data centers, which are small to medium-size (for instance, clusters) belonging smaller outfits like co-location companies, research labs, and educational institutions [1]. Therefore, the majority of data-center operations are unable to *fully* exploit LMP like their larger counterparts.

In this paper we describe a market-based framework to increase access to geo-diversification for *all* data centers. We envision a market in which any data center can participate in the free trading of energy through the exchange of computation. Participating data centers may buy and sell resource allocations from each other based on the current state of their respective energy costs. For this market to be successful in a real world environment, there are many factors that must be considered. The purpose of our research is to assuage any concerns via the construction of a simulation that mimics the day-to-day operations of such a market. Real workload traces inform our models on the inner workings of a data center, which includes considerations for workload distribution, arrival rates, I/O characteristics, data footprint, *etc.*, to how they translate to overall power consumption and costs. The creation of representative data-center models and the availability of real energy-pricing data together allow us to study the viability of an energy-trading market at various scales of deployment.

Specific contributions of this work include:

- An open market for trading jobs among data centers is proposed, which allows small and medium data centers to access geo-diversification.
- Our data center operations (workload and power usage) were modeled based on real workload traces and are consistent with previous research. Our models inform a novel algorithm for a data center to make decisions on whether to enter the market, and whether to buy or sell jobs for revenue or cost reduction, respectively.
- We conducted a simulated feasibility study and showed that, with enough data center participants, the market affords data centers significant increases in profit through energy-cost reductions and higher job completion rates.

The remainder of this paper is organized as follows. In Section II we present the system overview. Section III explains the methods used to simulate by market participants, forming realistic energy transactions. Experimental design and results are presented in Section IV. Section V outlines the related work, and we conclude our findings in Section VI.

II. Proposed Framework

The overarching framework to support an energy market is depicted in Figure 1. The framework is organized into two tiers of operation.

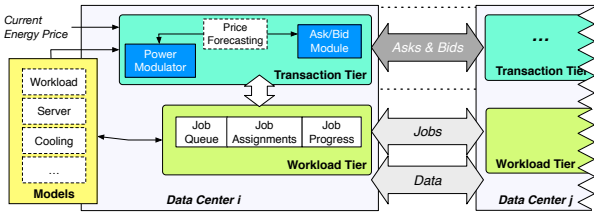

Fig. 1: Transactional Framework for Workload Exchange

Transaction Tier: At any given time, a data center may decide to either curtail or ramp-up energy usage. The Transaction Tier must consult the current energy prices (input), prediction models, workload states, and solve optimizations to make sound decisions that include, but may not be limited to, the size, duration, and the cost of the ask or bid for resources. The data center might be satisfied through: (1) adjusting its aggressiveness in scheduling local jobs, (2) buying additional jobs that are off-site to increase revenue, or (3) placing jobs for sale on the market. In the latter two cases, the Transaction Tier can submit the following types of requests to its peers:

- *Ask:* Requests computational resources in other data centers. An ask may include the number of servers, server specifications, start time, and a duration of the requested servers.
- *Bid:* A bid is an offering of local resources in response to an existing ask. Bids can also be unsolicited if the data center is eager to use up its cheap energy and available server capacity. Bids include the number of available servers, server specifications, start time, cost, and duration for which the prescribed costs are guaranteed. After the bid's duration has elapsed, the resources or costs may become volatile.
- *Accept & Commit:* Signals an acceptance of an ask or bid with a second party. Conversely, a commit acknowledges the acceptance from the other party, and the resources become available at the agreed-upon start time.
- *Cancel:* Cancels an existing ask or bid that has yet to be accepted.

Workload Tier: The purpose of this tier is two-fold. On one hand, it informs the Transaction Tier on making right-sized job purchases or sales, and on the other hand, it communicates with the Server Tier below to acknowledge its locally available resources. The components in this tier are therefore responsible for the monitoring, deferment, scheduling, and migration of workloads as means to maximize performance objectives such as response time. The current workload state is monitored, including: the job queue, each job's progress, and the servers onto which the jobs are assigned. The workload state provides input to the power models, which allows the Transaction Tier to make proper decisions in the market.

To operationalize the two tiers, we need to introduce some system models. We assume a data center comprises a set of *clusters* and that each cluster is assigned multiple servers on which tasks are executed. For simplicity, servers are assumed to be homogeneous, each with a predefined amount of CPU, memory, and disk capacity.

The data center's execution model is represented as a queuing system, in which jobs arrive at constant time intervals. We define a sequence of time units $(t_0, t_1, ..., t_{n-1})$, and adjacent time intervals $t_i - t_{i-1}$ are constant, for all i. A λ arrival rate is used to assign the average number of jobs that a single data center should observe in a given interval, and the actual number of jobs that arrive are based on a Poisson distribution. We introduce a balking rate based on the current stress of the data center, defined as the ratio between the combination of CPU, memory, and disk utilization at time t_i over the *total capacity* of those resources (*e.g.*, at 33% capacity, a third of jobs will balk).

Jobs and *tasks* are the subject of all of work done in data centers. A job J may consist of a set of tasks $J = \{j_0, j_1, ...\}$, and each task $j \in J$ represents a single schedulable and executable thread. A job J associates with a tuple $(cpu_J, mem_J, disk_J, rev_J, dur_J)$, where $cpu_J, mem_J, disk_J$ refer to J's normalized CPU utilization, RAM, and disk requirements. rev_J denotes the revenue generated for the completion of job J, and dur_J refers to J's predicted execution time given those prior resources. These values are divided by task, *i.e.*, $\sum_{j \in J} cpu_j = cpu_J$. Based on $disk_j$, we estimate its amount of input data ($dataIn_j$) and output data ($dataOut_j$). These designate the I/O costs associated with the job upon its arrival to the data center, and the total amount of data it will generate upon completion. We assume that a task's memory footprint grows or shrinks linearly, *i.e.*, adjusted by $\frac{dataOut_j - dataIn_j}{disk_j}$ per time unit. The current memory footprint for tasks is tracked, because it directly affects job transfer costs to a different data center.

III. Transacting on the Market

The market agent was designed with free market transaction in mind. We want a system which can assist data centers in keeping under budget while helping to maximize profit. We sought to populate the market with sellers (those offering jobs to offload), and buyers (those looking to purchase and finish execution of others' jobs). The largest hurdle to clear is to predict *when* work should be offloaded.

We assume that each data center is assigned a participation interval, τ. At the start of every τ time units, the data center predicts future costs and considers entering the market to increase profit. To project cost, we require the list of currently executing jobs, and the current total resource usage (stress) of the data center. Using these, we iterate through the list of currently executing jobs, calculating the cost and stress that each job is placing on the local data center, stopping when the cumulative stress of the jobs match the stress currently being placed on the center.

These jobs are stored in a list, which is then filtered for all jobs that would finish during the next τ interval, removing the stress of these from the stress total. We then repeat this process until there are no more jobs that would see execution during the interval. The total relative cost of all of these jobs is compiled and divided by the average failure rate for a single cluster. The result is then factored into the τ time interval, giving us a final total projected cost value. A data center will enter the market as a job-seller if the following condition is met:

$$p(t_{i+1}) > \frac{budget}{t_i - t_{i-1}} + \epsilon$$

where $p(t_{i+1})$ is the projected cost in for the next time unit, t_{i+1} and ϵ is a buffer put to prevent data centers from

entering the market if they are a negligible amount over or under budget. Otherwise, the data center will decide to enter the market as a buyer. Upon market entry, a seller can choose places to send work, and a buyer can offer a resource availability and a price.

Let us consider the case in which the energy rates for a data center is projected to increase over the next time interval, and that it has determined that selling jobs to a third party would allow it to maximize profit. Two problems arise: First, how does a data center determine *which* jobs would be most beneficial to offload? Second, how does a data center determine fair market prices the jobs?

Recall that each job J is associated with a vector $\langle cpu_J, mem_J, disk_J, rev_J, dur_J \rangle$. Assume that job J has been executing for T time units, where $0 \leq T \leq dur_J$ on the local data center. If J is to be offloaded to a remote site, we will assume that the revenue for completing the job will be transferred to that site. Therefore, the local data center would be interested in retaining only the proportion of revenue for processing it for T time units. We set the price of J on the market to be,

$$price_J = rev_J \left(1 - \frac{dur_J - T}{dur_J}\right) + migCost_J$$

$$migCost_J = \frac{(dataIn_J + dataOut_J)}{BW} \cdot transferCost$$

where $migCost_J$ is the overhead of migrating J to the remote data center. The cost of migration is nontrivial, as it must transfer the necessary input data $dataIn_J$ plus the intermediate output data $dataOut_J$ over a wide area network, whose average bandwidth BW determines the overall job transfer time. The transfer time is then factored into a cost per unit time to derive the total migration time for J.

We now describe the process of identifying candidate jobs to offload. The first objective we are seeking to minimize job failure rate by maximizing the total time a job has left to execute post-transfer. The other objective aims to maximize the number of low-revenue jobs sent out of the data center, $\sum \frac{1}{rev_J - cost_J}$. We seek to find the optimal set of jobs from the candidate list to send to the market. In order to include both factors, these individual candidate lists are created, normalized, weighted, and then finally combined before being sorted and selecting the final bundle.

Once it has been determined that a data center is under budget, all that is required to determine an onload is projected cost and the budget for the given time interval, $\frac{budget}{\tau}$. The theoretical maximum that this data center could take on while still remaining under budget is calculated, and the amount the center is currently taking is subtracted from this, giving the amount of buffer with which the data center has to operate. This value then multiplied by the maximum cost that could be incurred by the center, gives the estimated total cost of bringing new work into the center.

IV. Experimental Results

For a market to be viable, it must be mutually beneficial for the participants. In this section we present preliminary results. To test the validity of our market, we implemented a discrete-event simulator.

A. Experimental Setup

First we describe data sets and how system parameters were derived in our experimental setup.

1) Data Center Operation: We modeled our data-center operations based on the Google workload trace data [2], which contained information pertaining to job/task scheduling, including their resource allocation (*e.g.*, CPU, memory, disk) and duration. According to Alam, *et al.*, the trace data was taken from a single cluster containing of 11000 cores [3]. The actual resource usage numbers themselves were obfuscated by Google using a linear transformation, normalizing all values to $[0, 1]$, which makes it difficult to ascertain the exact hardware used in the trace. In our simulation we assumed that all CPUs are homogeneous and contained 10 cores. Under ideal conditions, each 10-core CPU can execute 1200 processes per minute on each core.

A *task* is a single-core process deployed for execution. We model tasks using the same method as proposed by Mishra, *et al.* [4]. They categorized length and type of tasks, breaking down average normalized requirements for each category. The categories are shown in Table I. Tasks are split upon creation:

Size	Core	RAM	Local Disk	Duration
S	0.2	0.2	0.0001	3 - 20 minutes
M	$0.3 - 0.51$	$0.3 - 0.5$	0.001	3 - 20 minutes
L	$0.5 - 1.0$	$0.5 - 1.0$	0.01	18 - 24 hours

TABLE I: Task Types

90% generated are Small, and the remaining 10% are Large. Jobs are classified as short (0-2 hours) at a 90% rate and long jobs (18 - 24 hours) at a 10% rate. According to Di, *et al.* [5], this rate is consistent with the split of jobs present in the Google cluster trace.

To assign a representative job λ-arrival rate for the model, we used numbers reported by Chen, *et al.* [6], which specified that 57.6% of jobs were completed, whereas 40.1% of jobs were killed by the system scheduler. We found that a λ rate of 0.13875 results in the same percentages for a single cluster in the simulation. Finally, we assume that simulated data centers are connected over wide-area networks with average bandwidth (BW) ranging anywhere from 0.5 Gbps to 2 Gbps. This rate is used for estimating job transfer times from site to site.

2) Power-Usage Costs: A data center's total power usage is largely dominated by two factors: overall server power consumption and a cooling overhead. To model the former, we assume the idle power consumption of a CPU to be 161W and the peak consumption was set to 230W. We assign the cooling overhead to be 33% of the total server consumption at any point in time; this rate is informed by previous research [7]. On the cost side, we acquired hourly energy-pricing data from Sep 1, 2017 to Sep 1, 2018 pertaining to five major power utilities covering a wide geographical expanse in the U.S.,

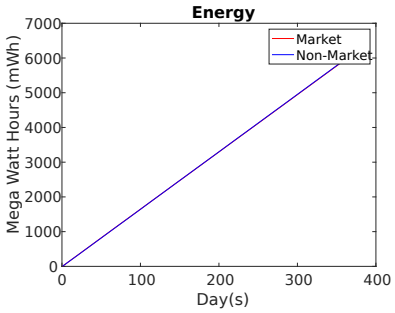

Fig. 2: Aggregated Energy Usage (2 Participants)

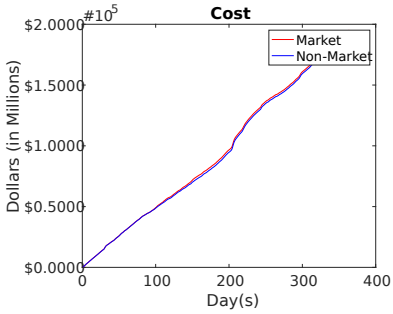

Fig. 3: Cumulative Cost (2 Participants)

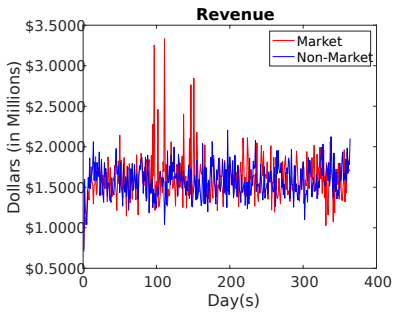

Fig. 4: Revenue (2 Participants)

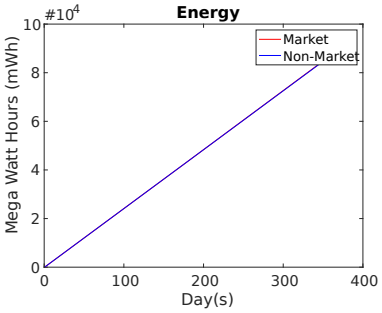

Fig. 5: Aggregated Energy Usage (100 Participants)

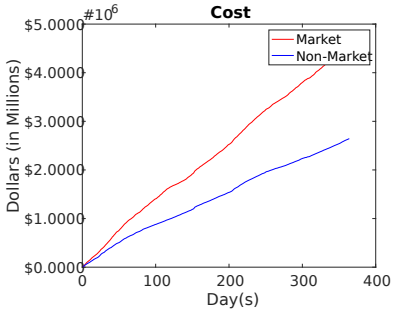

Fig. 6: Cumulative Cost (100 Participants)

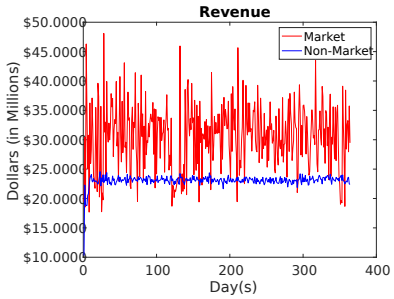

Fig. 7: Revenue (100 Participants)

namely CAISO, NYISO, SPPISO, MISO, PJM, covering 25 states and spanning all 4 time zones.

B. 2-Participant Results

In order to thoroughly vet the market, we sought to test the system under two different scenarios. In this first scenario, we are interested in showing the impact between two isolated data centers in opposing geographic locations in the U.S. The two ISOs we chose for the simulation were CAISO and NYISO. We placed a 2-cluster data center in CAISO, and a 5-cluster (medium-sized) data center in NYISO. Figure 2 shows the aggregated energy usage (mWh) for the two data centers when participating in the market (`market`) and when they are not (`non-market`) for a full year. There is very little to no discernible difference. In fact, the total energy used in the market was 600400 mWh, while the total used in the non-market scenario was 601040 mWh, accounting for only a net 0.07% decrease in energy saved.

The cost, shown in Figure 3 however, tells a slightly different story. Unlike energy usage, where we saw a minor reduction, the total cost for operating in the market actually increased slightly. The total cost of participating in the market for a year was $19,523,000$ while the total cost for operating without the market was $19,364,800$, resulting in a 1.25% increase in cost for operating in the market. The revenue does its best to make up for cost in this scenario, peaking at $83,903,000$ for the centers participating, where as those in the non-market peaked at $82,759,000$. In total, it resulted in a 1.91% increase in revenue for the market over the non-market.

We also observed metrics related to performance. In the `market` scheme, the job throughput dipped 0.89% compared to to `non-market`. We also saw a 0.9% increase in job failures, likely due to the migration times adding too much overhead for the jobs to meet their deadlines. Overall, if there is low data-center participation in the market, little is to be gained, and in fact, it may even prove costlier to data centers.

C. 100-Participant Results

Next, we simulated the market with a higher number of participants. We distributed 100 mixed-size data centers across the U.S. based on geographical distributed provided by [8], there are 125 data centers in NYISO's operating region, 181 in CAISO, 220 in MISO, 263 in PJM, and 58 in SPP. Using these distributions, we scaled the number of data centers down to 100, placing 15 in NYISO, 22 in CAISO, 26 in MISO, 31 in PJM, and 6 in SPP. The results of this simulation were as follows.

Figure 5 shows the total energy usage for all 100 data centers participating in the workload exchange market versus the non-market scheme. Much like the small run from earlier we see very little change in the overall energy usage. The market used $8,819,300$ mWh total while the non-market used $8,805,100$ mWh total. This accounts for a small increase in overall energy usage (0.09%).

In Figure 6, we can observe a much more pronounced increase in overall energy costs. This is expected; with there being more activity on the market, more jobs are being executed over the same period of time (this is later verified

with results showing higher job throughput). The total cost of operating in the market peaked at $452,590,000$, while the non-market peaked at $264,420,000$, a significant increase of 65.49%.

Figures 7 reports the revenue for each time instant. The market action is quite strong, suggesting that many jobs are being traded. The cumulative revenue shows that in total, the market actions lead to significant gains in profit. The market saw a peak revenue of $1,598,500,000$, while the non-market peaked at $1,206,700,000$. Subtracting the (higher) costs from these figures to obtain overall profit, we observe a 17.8% increase in profit when using the market.

V. Related Work

Larger-scale data centers can be geographically distributed over a wide-area network, and if the participating data centers are located in different energy markets, then opportunities for cost reduction can be exploited. Several authors propose the *follow-the-renewables* policy [9]–[12], where workloads are routed among various green data centers to take advantage of their local renewables. Geographical load balancing focuses on shifting workload to locations with lower energy prices. Qureshi, *et al.* present an analysis of data centers' cost reduction by simulating traffic routing to various data centers in wholesale energy markets [13]. Buchbinder, *et al.* propose online solutions for migrating batch jobs to optimizing costs [14].

Rao, *et al.* minimize overall costs by solving for optimal resource allocation and request rates at multiple data centers [15], [16]. Chen, *et al.* presented a centralized scheduler that migrates workloads across data centers in a manner that minimizes brown energy consumption while ensuring the jobs' timeliness [17]. Zhang, *et al.* additionally consider meeting a budget cap [18]. Liu, *et al.*'s work on greening geographical load balancing [19], [20] assumes a general Internet service-request workload for data centers located in various geographical regions. They proposed distributed algorithms for minimizing aggregated costs by solving for an optimal number of active servers per data center and a load balancing policy (request routing). Adnan, *et al.* consider online optimization of job schedules, then using migration to reconcile prediction errors for optimizing costs while meeting deadlines [21].

VI. Conclusion and Future Work

In this paper we proposed an transactional energy market in which *any* data center can trade jobs across geographical locations to exploit differences in energy pricing. We modeled our system using real workload traces, and acquired actual energy-pricing data to conduct our feasibility study. Overall, we showed that, through market participation, data centers can substantially increase profit and job throughput.

With the feasibility study completed, our future work involves providing the models and mechanisms for data centers to trade jobs given dynamic pricing signals. One area of focus will be on the migration of common job types. Novel work-transfer and scheduling mechanisms must be developed as part of the supporting framework to transparently operationalize transactions over heterogeneous parallel architectures.

References

[1] P. Delforge, "Data Center Efficiency Assessment," tech. rep., Natural Resources Defense Council (NRDC), 2014.

[2] Google, "google/cluster-data."

[3] M. Alam, K. A. Shakil, and S. Sethi, "Analysis and clustering of workload in google cluster trace based on resource usage," *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, 2016.

[4] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das, "Towards characterizing cloud backend workloads," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 4, p. 34, 2010.

[5] S. Di, D. Kondo, and W. Cirne, "Characterization and comparison of cloud versus grid workloads," *2012 IEEE International Conference on Cluster Computing*, 2012.

[6] X. Chen, C.-D. Lu, and K. Pattabiraman, "Failure analysis of jobs in compute clouds: A google cluster case study," *2014 IEEE 25th International Symposium on Software Reliability Engineering*, 2014.

[7] U. Hoelzle and L. A. Barroso, *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 1st ed., 2009.

[8] "https://www.datacentermap.com/."

[9] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Free lunch: Exploiting renewable energy for computing," in *Proceedings of the 13th USENIX Conference on Hot Topics in Operating Systems*, HotOS'13, (Berkeley, CA, USA), pp. 17–17, USENIX Association, 2011.

[10] D. Chiu, C. Stewart, and B. McManus, "Electric grid balancing through low-cost workload migration," *ACM Sigmetrics Performance Evaluation Review (GreenMetrics'12)*, 2012.

[11] Y. Li, D. Chiu, C. Liu, L. T. Phan, T. Gill, S. Aggrawal, Z. Zhang, B. T. Loo, D. Maier, and B. McManus, "Towards dynamic pricing-based collaborative optimizations for green data centers," *2nd International Workshop on Data Management in the Cloud (DMC'13). Held in conjunction with ICDE.*, 2013.

[12] C. Johnson and D. Chiu, "Hadoop in flight: Migrating live mapreduce jobs for power-shifting data centers," *9th IEEE International Conference on Cloud Computing (CLOUD'16)*, 2016.

[13] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 123–134, Aug. 2009.

[14] N. Buchbinder, N. Jain, and I. Menache, "Online job-migration for reducing the electricity bill in the cloud," in *Proceedings of the 10th international IFIP TC 6 conference on Networking - Volume Part I*, NETWORKING'11, (Berlin, Heidelberg), pp. 172–185, Springer-Verlag, 2011.

[15] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment," in *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM'10)*, pp. 1145–1153, 2010.

[16] L. Rao, X. Liu, M. D. Ilic, and J. Liu, "Distributed coordination of internet data centers under multiregional electricity markets.," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 269–282, 2012.

[17] C. Chen, B. He, and X. Tang, "Green-aware workload scheduling in geographically distributed data centers," in *CLOUDCOM*, 2012.

[18] Y. Zhang, Y. Wang, and X. Wang, "Electricity bill capping for cloud-scale data centers that impact the power markets," in *ICPP*, pp. 440–449, 2012.

[19] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew, "Greening geographical load balancing," in *SIGMETRICS'11*, pp. 233–244, ACM, 2011.

[20] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew, "Online algorithms for geographical load balancing," in *Proc. Int. Green Computing Conf.*, (San Jose, CA), 5-8 Jun 2012.

[21] M. A. Adnan, R. Sugihara, and R. K. Gupta, "Energy efficient geographical load balancing via dynamic deferral of workload," in *IEEE CLOUD*, pp. 188–195, 2012.