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—Data centers are among the largest users of electricity. To curtail costs, an organization might place associated data centers in various geographical regions to exploit multiple energy markets that offer different pricing. This is process is known as geo-diversification. However, due to large capital expenditures involved, geo-diversification has so far only been feasible for major data-center operations (e.g., AWS). However, studies show that over half of the data centers in existence today are small to medium-sized “clusters” independently operated by local businesses, labs, and academic institutions, which lack the capital resources required to geo-diversify. Therefore, most data centers are locked into their existing locations and cannot exploit the broader energy market to reduce costs.

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 an average increase of 17.8% profit margins, 12% increase in total job throughput, and a 10% reduction of job failures.

I. INTRODUCTION

For over a century, the power grid has driven the national and global economies. To sustain its use by future generations, the rate of deploying renewable power sources (e.g., windmills and solar farms) has increased exponentially. However, incorporating substantial amounts of renewables into the existing grid infrastructure has been slow, costly, and even detrimental to the environment [1]. This problem stems from many renewable resources’ stochastic availability and the absence of inexpensive large-scale energy storage [2]–[9].

One way to manage intermittent power supply is to regulate load (or demand) to match the availability of renewables. To this end, utilities have introduced demand response programs through locational marginal pricing (LMP), the current model for electricity pricing across the US [10]–[12]. Specifically, utilities base realtime prices on the hypothetical incremental cost to the system of redistributing 1 kWh of energy from the current pool of resources available to it. With the amount of daily active users of energy present, that hypothetical cost is often driven quite high, which heightens the cost to any constant consumer of energy, with the most notable effects of this made present in the most prodigious users of energy: data centers.

Spurred on by growing demand in the tech sector, data centers have been revealed as being among the heaviest users of electricity [13]–[15]. Not surprisingly, research on curtailing data centers’ energy consumption has grown accordingly in recent years, and energy-efficient advancements now permeate modern data-center design. Although these efforts have had a positive effect toward improving data-center efficiency, data centers’ aggregated energy usage is still projected to grow at a 7% annualized rate to year 2020, equating to 140 billion kilowatt-hours per year (kWh) [16]. Put into perspective, an average data center consumes electricity at a rate of 20 MW, equivalent to roughly 16000 U.S. households [17], [18].

To minimize total operating costs, major data-center operations have begun exploiting the variability of energy price signals. For instance, a data center could adapt their workload scheduling to consider its local LMP, but it is at the risk of reduced job throughput. A more ambitious approach to exploit LMP variability is through geo-diversification, in which an organization builds multiple data center locations across disparate geographical regions. The economics of geo-diversification is sound: when energy is cheap at a certain location, computational workloads can be scheduled more aggressively and even migrated there from locations experiencing higher prices. However, due to the tremendous 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 [19]. 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 improve 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 computational resources. 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 allay any concerns via the construction of mathematical models that mimic the day-to-day operations of a data center. 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.

Our results show that when operating our market agent increases profit, and jobs completed, while reducing jobs failed. All of this is done with little to no effect on the total energy used, even decreasing the total energy used in the same amount of time by a small margin. We have also found that the results of the market scale with the overall participation, which is to say the more participants in the market, the more beneficial the market is to all involved.

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 reduction and higher job completion rates.

The remainder of this paper is organized as follows. In Section II we present the system overview. Section III and Section IV explain the models and algorithms performed by market participants to form valid and productive energy transactions. Experimental design and results are presented in Section V. Section VI outlines the related work, and we conclude our findings in Section VII.

II. PROPOSED FRAMEWORK

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

Transaction Tier: Given the current energy prices and a price-prediction model, 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/bid. In either case, the data center might be satisfied through: (1) adjusting its aggressiveness in scheduling local jobs, (2) buy additional jobs that are off-site to increase revenue, or (3) placing jobs for sale on the market.

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.

To inform the Transaction Tier, 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 a proper decisions in the market.

III. MODELING OPERATION AND COSTS

In our paper we assume a data center comprises a set of clusters. Each cluster is assigned multiple servers on which tasks are executed. For simplicity, while a data center may contain varying numbers of clusters and servers per cluster, the servers themselves are assumed to be homogeneous, each with a fixed 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. Let us define a sequence of time units as \((t_0, t_1, ..., t_{n-1})\), and adjacent time intervals \(t_i - t_{i-1}\) are constant, for all \(i\). A \(\lambda\) 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 \(t\) over the total capacity of those resources. At 33% capacity, \(\frac{1}{3}\) of jobs will balk, at 66%, \(\frac{2}{3}\) will balk, and at 99% all jobs will balk.

Jobs and tasks are the subject of all of work done in data centers. A job \(j\) is defined as a set of tasks \(J = \{j_0, j_1, ...\}\). A task \(j \in J\) represents a single executable process, and each task is associated with a tuple \((cpu_j, mem_j, disk_j, dur_j)\), where \(cpu_j\), \(mem_j\), \(disk_j\), refer to task \(j\)’s CPU utilization, RAM, and disk requirements, and \(dur_j\) specifies \(j\)’s expected execution time given those prior resources. 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 \((\text{dataOut}_j - \text{dataIn}_j)\) per time unit. The current memory footprint for tasks is tracked, because it directly affects job transfer costs to a different data center.

A task is assigned for execution on one of the data center’s \(m\) homogeneous CPUs. For each time unit, the data center traverses through each of its clusters and determines whether any job has completed. After reclaiming the resources that were allocated to completed jobs, the system then schedules jobs waiting in the queue. If we define CPU assignment,

\[
a_{ij}(t) = \begin{cases} 
1, & \text{if task } j \text{ is executing on CPU } i \text{ at time } t \\
0, & \text{otherwise}
\end{cases}
\]

then the power consumption of the \(i\)th CPU can be estimated to be,

\[
u_i(t) = \sum_{j=0}^{n-1} a_{ij}(t) \cdot cpu_j
\]

where \(cpu_j\) is task \(j\)’s CPU utilization, as given before. The data center’s total power consumption at time \(t\) can therefore be defined as follows,

\[
power(t) = \sum_{i=0}^{m-1} \left( u_i(t) \cdot W_{\text{max}} + (1 - u_i(t)) \cdot W_{\text{idle}} \right) + \sigma(t)
\]

where \(W_{\text{max}}\) and \(W_{\text{idle}}\) refer to the CPU’s power consumption at peak utilization and when idling, respectively, and \(\sigma(t)\) is the cooling overhead required at time \(t\).

The total cost of operation for a data center located in geographical region \(r\) is aggregated over time,

\[
\text{cost}(r, t) = \sum_{t=0}^{n-1} \text{rate}(r, t) \cdot \text{power}(t)
\]

where and \(\text{rate}(r, t)\) refers to the energy rate in region \(r\) at time \(t\).

IV. Transacting on the Market

The market agent was designed with a free market transaction in mind. We want a system which can assist a data center in keeping under budget while helping to maximize profit for all data centers involved. 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 might be offloaded.

A. Point of Inception

We assume each data center is assigned a participation interval, \(\tau\). At the start of every \(\tau\) time units, the data center predicts future costs and considers entering the market to increase profit. To project cost in our model, we require the list of currently execution jobs, and the current total resource usage (stress) of the data center. Using these, we index through the list of currently executing jobs, calculating the cost and stress that each job is placing on the local data center, and stopping when the total 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 any jobs which would finish during the next \(\tau\) 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. Finally the cooling cost \(\sigma(t)\) is added on, and the resulting number is factored into the \(\tau\) time interval, giving us a final total projected cost value. Finally, to determine whether a data center enters the market as a buyer or seller. A data center will enter the market as a job-seller if the following condition is met:

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

where \(p(t_{i+1})\) is the projected cost in for the next time unit, \(t_{i+1}\) and \(\epsilon\) 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 of jobs. Upon market entry, a job-seller can choose places to send work, and a buyer can offer a resource availability and a price.

B. Selling Jobs

Let us consider the case in which the energy rates for a data center is projected to increase in the next \(\tau\) time interval, and that it has determined that selling (even currently executing)
jobs to a third party would be most the beneficial to its profit, allowing it to lower its power usage. Two problems arise: (1) First, how does a data center determine which jobs would be most beneficial to offload? (2) Second, how does a data center determine fair market prices for the jobs?

Recall that each job $J$ is associated with a vector $(cpu, mem, disk, rev, dur)$. Assume that job $J$ has been execution 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 returning 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 $J$.

Algorithm 1 describes the process of identifying candidate jobs to offload. The first we are seeking to minimize job failure rate by maximizing the total time a job has left to execute post transfer, represented by $\sum \text{slack}_j - mig_j$. The other is aiming to maximize the number of low revenue jobs sent out of the data center, represented by $\sum \text{rev}_j - \text{cost}_j$. In order for a data center to set their preference to which they want to maximize over, we introduced a constant $\alpha$ as a weighting value for each of the individual output vectors of the separate algorithms. This algorithm seeks to find the optimal bundles 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 sorting and selecting the final bundle.

While most work in this algorithm is trivial, the single sort required makes the overall run time of the algorithm $O(n \log n)$. Once the bundle of choice has been selected, those jobs are then sent to the method $setMu$ which generates a binary vector. This vector represents the list of all in-progress jobs, where a 1 represents a job that is available for sale, with a 0 in place otherwise. Finally a tuple is generated for the data center of the form:

$\langle \text{Core Space, RAM, Local Disk, Cost to data center} \rangle$

Finally, this seller enters the market.

C. Buying Jobs

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, $\frac{\text{budget}}{t}$. 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, giving us the total cost to the data center of bringing new work into the center. A tuple of the form:

$\langle \text{Core Space, RAM, Local Disk, Cost to data center} \rangle$

is generated here as well. It is used to aide in determining where to make transactions.

V. EXPERIMENTAL RESULTS

In this section we present our experimental results. To test the validity of our market, we implemented a discrete-event
simulator. We simulated an open market for varying degrees of data-center participation. For a market to be viable, it needs to be universally beneficial. It should not solely benefit the data centers participating in the market, likewise it should not strictly benefit the customers using these centers to complete jobs. The main goal of this market system is to reduce cost and energy usage, while increasing the flow of jobs through a distributed market system. In order for us to consider our market a success in these tests we would hope to see marked improvements to job completion without major sacrifices to revenue, energy usage, or job failure rates.

A. Experimental Setup

Here 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 [20], 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 [21]. The actual numbers themselves were been 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.

A task is a single-core process deployed for execution. We model tasks using the same method as proposed by Mishra, et al. [22]. 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:

<table>
<thead>
<tr>
<th>Size</th>
<th>Core</th>
<th>RAM</th>
<th>Local Disk</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>0.2</td>
<td>0.2</td>
<td>0.0001</td>
<td>3 - 20 minutes</td>
</tr>
<tr>
<td>M</td>
<td>0.3 - 0.5</td>
<td>0.3 - 0.5</td>
<td>0.001</td>
<td>3 - 20 minutes</td>
</tr>
<tr>
<td>L</td>
<td>0.5 - 1.0</td>
<td>0.5 - 1.0</td>
<td>0.01</td>
<td>16 - 24 hours</td>
</tr>
</tbody>
</table>

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. [23], this rate is consistent with the split of jobs present in the Google cluster trace.

To assign a representative job $\lambda$-arrival rate for the model, we used numbers reported by Chen, et al. [24], which specified that 57.6% of jobs were completed, whereas 40.1% of jobs were killed by the system scheduler. We found that a $\lambda$ 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: As noted in previous studies [17], the 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 ($W_{idle}$) to be 161 Watts and the peak consumption ($W_{max}$) was set to 230 Watts. We assign the cooling overhead ($\sigma$) to be 33% of the total server consumption at any point in time; this rate is informed by previous research [17].

To simulate the economics for a data center, we assigned a revenue ($rev_j$) for the completion of each job $J$ to be $rev_j = \max_{j \in J} dur_j \cdot \sigma \cdot$0.1. That is, $J$’s revenue is the product of a given rate and the longest running task in $J$. On the cost side, we acquired hourly energy-pricing data from Sep 1, 2017 to Sep 1, 2018 pertaining to five major ISOs covering a wide geographical expanse in the U.S., namely CAISO, NYISO, SPPISO, MISO, PJM. These ISOs service a set of corresponding states:

- CAISO - CA
- NYISO - NY
- SPP - OK, KS, NE, SD, ND
- PJM - OH, KY, WV, VA, PA, NJ, DE, MD
- MISO - MN, IA, MT, AR, LA, MS, IL, IN, WI, MI

totaling 25 states, spanning all 4 time zones (PST, MST, CST, EST). Time was tracked according to Coordinated Universal Time (UTC), translating it to the designated time zone for a given state. Each state contains and monitors the data centers that are existing and operating within their borders.

B. Two-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 discernable 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. 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 annual 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 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, these results suggest that little is to be gained, and in fact, it may even prove costlier to data centers.

C. Large-Scale Participant Results

Next, we simulated the market a higher number of participants. To obtain an accurate real-world portrayal, we distributed 100 mixed-size data centers across the U.S. based on geographical distributed provided. According to information gathered from [25], 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 because due to there being more activity on the market, which means more jobs are being executed over the same period of time (this is later verified with results showing higher job throughput). The increase is significant, however. The total cost of operating
in the market peaked at 452,590,000, while the non-market peaked at 264,420,000. This accounts for a large increase, at 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, shown in Figure 8, 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.

Figure 10 shows the result for job throughput. Unlike the earlier run for two participating data centers, total job completion rates observed a large increase with 100 participants. The total number of jobs completed in the market was 3,012,400,000, while the non-market only completed 2,699,400,000, resulting in an overall increase of 12.01%.

Similarly, the total number of jobs that failed decreased with market participation, as shown in Figure 10. This meant that some jobs that could not have been scheduled due to a lack of resource capacity were able to be completed when the data center offered them for sale to a remote site. This resulted in a 10.7% decrease in job failures. The overall profit increase is therefore explained by the relatively small increase in cost against the higher volume of jobs completed.

D. Discussion

The salient takeaway from these results is that the number of participating data centers matters. When the participation level is low, we observed only small hints that the market is in operation, however, overall the difference made is slight or counterproductive. The only place where there is a noticeable divergence in total usage is revenue. For most every other metric, the market and non-market systems stay even with each other over the entire year period. This is an important result, as this shows that the system is balanced and will not attempt to take advantage of different sized clusters. The market is designed so that some given amount of space on a small cluster is not weighted the same as the same amount of space on a large cluster. With a larger cluster and a smaller cluster using the system in harmony, this shows promise for allowing the market to execute in a data center rich environment.

The results for the 100 data center run are a very clear depiction of what this market would look like given more free reign to operate with a large and geo-diverse client base. To be clear, if our market is to be considered a success, we want to see improvement to job throughput, without major sacrifices to job failures, energy, or overall profit. We showed that, interesting, like the low-participation simulation, the market shows little to no change in total energy usage. However, beyond that, the differences are far more pronounced.

First, the cost has increased sharply. This, however, is expected. An increase in cost is only an indication that more energy is being used up. It is not, by itself, a detriment to the market approach. The increases in profit, job throughput and reductions in failures easily offsets the costs, often doubling, or more than doubling the controlled non-market revenue. Finally, and perhaps most important are the markets effects on the completion and failure rates of jobs. Not only did the rate of job completion increase by 15%, but the rate of job failures decreased by a similar rate, all while energy usage has not changed from the controlled environment.

VI. RELATED WORK

It has been well-established that electricity is the dominant cost of modern data-center operations [16], [17], [19], [26]. Because data centers have a mutual interest to curb electricity costs, they can have a significant impact when participating in DR programs [27]. With technologies allowing flexible power management, the class of green data centers have emerged that exploit electricity costs. Early efforts consider co-locating renewable sources, e.g., by tying solar panels and wind turbines into data center architectures [28]–[31]. GreenSlot [32] and GreenHadoop [33] are two related systems that schedule jobs to match the availability of onsite-solar energy, while meeting deadlines. In GreenHadoop, the authors use general models to predict solar energy availability and the energy costs of Hadoop jobs, which are then used to inform resource allocation. Blink is another intermittent-energy aware management algorithm that leverages fast active/inactive states [30]. Liu, et al. consider batch job scheduling according to dynamic energy prices, energy storage, cooling options, and the availability of a local solar supply [34].

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, 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 [35]. Buchbinder, et al. propose online solutions for migrating batch jobs to optimizing costs [36].

Rao, et al. minimize overall costs by solving for optimal resource allocation and request rates at multiple data centers [37], [38]. 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 [39]. Zhang, et al. additionally consider meeting a budget cap [40]. Liu, et al.’s work on greening geographical load balancing [41], [42] 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 [43].
VII. CONCLUSION AND FUTURE WORK

In this paper we proposed an transactional energy market in which any data center can participate 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

