Towards Dynamic Pricing-Based Collaborative Optimizations for Green Data Centers

Yang Li† David Chiu‡ Changbin Liu* Linh T.X. Phan† Tanveer Gill†
Sanchit Aggarwal† Zhuoyao Zhang† Boon Thau Loo† David Maier§ Bart McManus**

University of Pennsylvania† Washington State University‡ AT&T Labs - Research*
Portland State University§ National University of Singapore§ Bonneville Power Administration**

***Abstract*—Increased demand for cloud computing services has ushered power management schemes into the frontlines of data center research. Meanwhile, market penetration of intermittent renewable energy sources (*e.g.*, wind and solar) is on the rise. While clean and abundant, their intermittency is troubling for utility companies, requiring power balancing reserves to be deployed at anytime to precisely match consumer demand with energy availability. However, a transformative redesign of our power grid is looming, calling for the use of dynamic energy pricing to resolve this issue by possibly shaping demand. Data centers, being significant consumers with the ability to adjust power utilization in real-time (*e.g.*, by migrating its jobs to and from other locations), are ideal candidates to participate in dynamic pricing markets.**

We propose a collaborative cost optimization framework by coupling utilities with data centers via dynamic pricing. We develop models describing the information exchange framework for utilities and data centers and employ a distributed constraint optimization solver, *Cologne,* to negotiate a mutually optimal price. An evaluation of our system has been performed using real intermittent-energy-generation trace data. Modeling the dynamic price over this trace, we show that our technique could reduce a participating data center's costs by 75%. On the side of utilities, we further show that consumer power demand can be shaped to reveal a 17% improvement on average.

I. Introduction

The computing flexibility afforded by Infrastructure-as-a-Service (IaaS) has accelerated the adoption of cloud-based solutions. In order to address this growing demand for computing services, data center operations have scaled in both power density and geographical presence. In fact, some large-scale data centers even replicate services across multiple distant locations to ensure availability [1]. To generate a stable revenue stream, data centers must therefore redouble their efforts towards reducing total operational costs. These costs, however, are a direct consequence of energy consumption, which has grown at an alarming rate over the past several years [2], [3], [4]. Indeed, studies as recent as 2011 have shown that today's data centers consume roughly 2.2% to 3.5% of total U.S. electricity use [4].

Meanwhile, a sea change in our nation's power distribution network should not be ignored by data center operations. The next-generation *Smart Grid* proposes to combat the current electrical grid's inefficiencies [5]. Chief among these inefficiencies is the absence of inexpensive large-scale energy storage, forcing electricity to be generated and consumed at the same rate. This unfortunate property requires *precise* amounts of electricity to be generated (or shed) at all times to match the customers' load. An imbalance in the form of either surplus or deficiency can overload grid components, culminating in service disruptions and even major outages. The lack of storage also hinders the rate of green energy (*e.g.*, wind and solar) penetration because their intermittency cannot be easily regulated or predicted. As a result, utility operators must face the volatility of increasing green energy integration by continuously deploying fast-reacting power reserves to maintain grid balance – currently a necessary overhead.

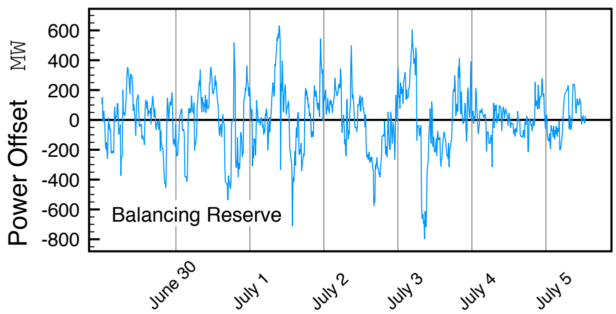
Improving energy efficiency is therefore a shared problem between both electric utilities and data centers. Data centers have the unique properties of being (1) major energy consumers with malleable and migratable workloads and (2) geographically distributed across different energy generation sources (e.g., Amazon Web Services has a multitude of data centers across continents). We envision a unification framework between data centers and electric utilities to enable mutual cost reductions. To realize such a framework, we assume that utilities can transfer dynamic real-time pricing (RTP) signals to their customers based on energy availability and demand (load) [6], [5]. In response, a geographically distributed data center could scale up or down energy usage by migrating large units of work to and from distant locations [7], [8], [9].

However, the negotiation of workload migration among multiple data center locations, each drawing power from a different electric utility, is nontrivial. In this paper, we propose the following vision and contributions:

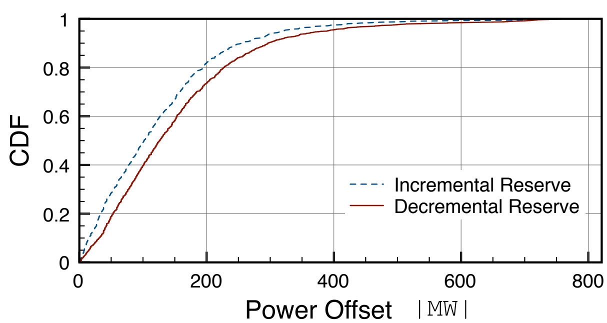
- A dynamic pricing negotiation mechanism to determine energy use within a data center such that electric utilities can ensure that its energy is being used effectively.
- To realize the dynamic energy pricing, we propose a constraint optimization formulation concerning multiple data center locations. This optimization problem executes on the *Cologne* [10] platform, which solves both centralized and distributed constraint optimizations using a declarative language. Each data center will use real-time pricing information to dynamically adjust its own energy usage via workload migrations (Section V). Our joint-optimization solution optimizes costs at utilities and data centers simultaneously.
- We evaluate the feasibility and applicability of our framework in a real world setting, using actual energy load and generation data sets from a major utility, Bonneville Power Administration (Section VI).

II. Motivation

The renewable energy market abounds in the Pacific Northwest region of the U.S., which has spawned the construction of many large-scale data centers in the area. The power utilities that support these data centers integrate a heterogeneous set of renewable resources. For instance, the Bonneville Power Administration (BPA, `www.bpa.gov`) has a set of 30+ hydroelectric dams, thermal plants, and wind fleets across four states. Recently, BPA announced it had integrated 4711 MW of *nameplate* (max capacity) wind energy. However, due to wind's intermittency, BPA must balance the deficiency or surplus stemming from imperfect prediction models in real-time using a fleet of ten dedicated hydroelectric dams, *i.e.*, its *balancing reserves*. To support this operation, BPA charges the connected wind farms an integration cost, inherently resulting in higher prices for wind energy [11].



(a) Surplus or Deficiency Power Offset through BPA's Balancing Reserve Deployments over 1-Week in July 2012



(b) Cumulative Distribution Function (CDF) of the Power Offset over the Same 1-Week Period

Fig. 1. BPA's Balancing (Incremental/Decremental) Reserve Deployment

BPA's grid operators must quickly ramp up generation from their incremental hydro balancing reserves to match consumer load to reconcile an unexpected wind deficiency. On the other hand, to compensate for excessive wind energy, operators must shed the surplus through its decremental balancing reserves, *e.g.*, water can be spilled over the hydroelectric dams instead of being passed through the turbines. While spilling has been a practical decrementing strategy, it is far from optimal: Wind energy would be offsetting the already green hydroelectricity, and spilling could further induce *gas bubble trauma* on protected fish populations [12].

We have analyzed BPA's balancing reserve deployment over a 1-week period in July 2012, which is shown in Figure 1(a). Observations are acquired at 5-minute intervals, and the oscillations highlight the complicated relationship between matching demand and green energy generation. We observe that incremental and decremental reserves were deployed 808 times and 1883 times, respectively. In further analysis displayed in Figure 1(b), we plot the cumulative distribution function (CDF) over the amount of power $|MW|$ balanced for each type of deployment. We can see that their distributions are quite skewed. For example, $|100|$ MW balancing constitutes half of the deployments. The higher frequency of decremental deployment also translates to wasted excess energy, resulting in abundant opportunities for data centers to utilize this surplus, if offered by the utility at a lower price.

We observe that while energy distribution may experience a localized skew (as shown in BPA's experience), it is likely the case that across geographic regions, the overall energy distribution stays fairly constant over time. Assuming that a fraction of power can be manipulated at each data center, then collectively, we can curtail a substantial amount of balancing reserve deployment. For instance, if excess wind energy is generated at one utility, then it could dynamically lower energy prices to local data centers in lieu of deploying decremental reserves. Similarly, if the wind subsides or if consumer demand increases unexpectedly, then an increase in energy prices may incentivize the local data centers to offload some power usage to other data centers.

In their 2009 work, Höelzle and Barroso estimated that a typical data center draws approximately 20 MW of power [1]. Taking into consideration that data centers can only control a fraction of its energy intake, they may not initially seem capable of curtailing balancing reserve deployments. However, in regions such as the Pacific Northwest, a high concentration of massive-scale data centers (*e.g.*, Google Dalles, Facebook Prineville, Amazon, and Yahoo!) has been constructed to take advantage of the abundance of green energy. These data centers are much larger than average. For instance, one building at Facebook Prineville consumes 28 MW, and it is projected to require a total of 78 MW to support all three planned buildings [13]. Google Dalles has also been estimated to consume anywhere from 50 MW to 103 MW [14]. As data centers continue to grow in power density, we believe they can cooperate with utilities to offset a significant percentage of balancing reserve deployments.

Towards dynamic pricing: In order to adjust demands based on energy availablity, we propose the use of a dynamic pricing mechanism that requires us to model the expected demand shifts of data centers whenever they are subjected to a new price from the utility company. While the high-level idea is straightforward, realizing the pricing model for both the utility companies and data centers is nontrivial. In particular, the following challenges need to be addressed:

- **Workload dependencies.** Essentially, the energy consumption of each data center is related to its workload. One common way of shifting the workload is to migrate jobs between locations, which is limited by a few factors. First, each data center might have a certain capacity that limits the number of jobs running on it. For example, in Hadoop, this capacity is better presented as the number of available map/reduce slots. Second, to perform a migration, a data center would encounter extra energy cost caused by, for instance, data transfer. Third, each job might have confidential restrictions that makes them trust only a subset of the data centers. All these factors increase the complexity of the model.

- **Price-demand shifts.** When modeling the action of a utility, a tiny price change might not be sufficient to trigger the data centers to make the expected energy shift. On the other hand, a significant price change might be too aggressive and lead to overheads triggered by load shifts across data centers. Therefore, a proper scheme to generate a new price is not straightforward to achieve.

III. Related Work

Several related works address intermittent energy integration within data centers. For instance, Stewart and Shen consider data center architectures that are equipped with onsite intermittent renewable sources [15]. They propose fine grained request-driven workload profiling and scheduling to maximize the use of available renewable energy. Goiri, *et al.* propose a number of scheduling techniques, such as *GreenSlot* [16] and *GreenHadoop* [17], which aim to maximize the total solar energy consumption by scheduling jobs based on a prediction on the available energy. Liu, *et al.* study workload and cooling management for a data center collocated with a solar microgrid [18]. They evaluate their optimization models over interactive and batch jobs and showed their scheduling algorithm can reduce non-renewable energy costs. These efforts, however, focus on integrating green energy sources that are collocated within the data center.

Green energy-efficient scheduling techniques for distributed data centers have been previously investigated. For instance, Le, *et al.* [19] propose a request distribution policy for Internet services to minimize the brown energy consumption. Similarly, Chen, *et al.* [20] presented a centralized scheduler that migrates workloads across data centers in a manner that minimizes brown energy consumption while ensuring the jobs' timeliness. Existing work in this direction, however, aims at optimizing the energy consumption from the consumer perspective only.

Several related efforts address data center's cost reduction through energy pricing. In *Blink* [21], the authors employ rapid active/inactive state switching (via PowerNap [22]) over a set of servers to adapt to variable power constraints. Rao, *et al.* seek to minimize overall costs for multiple data centers located in disparate energy marketing regions [23]. Akoush *et al.'s Free Lunch* architecture for cloud data centers shares several aspects of our goals [8]. The authors argue for either pausing virtual machine (VM) executions or migrating VMs between sites based on local and remote energy availability. Differing from prior work, our effort considers workload migration in a distributed data center environment for power adaptation, and moreover supports a coupling with utilities to mutually negotiate optimal prices.

In Liu, *et al.*'s geographical load balancing [9], the authors propose distributed algorithms for minimizing aggregated costs by solving for an optimal number of active servers per data center and a load balancing policy. HP Labs recently revealed the Net-Zero Energy Data Center, which can shape its demand based on supply [24]. Aikema, *et al.* studied the feasibility of using data centers as ancillary services (e.g., reserves) [25]. Chiu, *et al.* argue for an integration of utilities and data-center operations by showing mutual cost benefits given low-cost workload migration [26].

Moreover, Qureshi *et al.* [27] present a trace-driven analysis of the energy cost that data centers can save through distributed energy market pricing. The simulation in [27] used a heuristic traffic routing scheme that maps workloads to data centers with the cheapest electric price. Our research also advocates for promoting distributed energy market pricing. However, we further propose closely integrating data centers and electric utilities via a closed-loop interaction to achieve a dynamic pricing scheme and a load distribution that optimize both parties' objectives.

IV. Overview

Figure 2 presents an overview of our proposed system. The system is designed for a cloud environment consisting of multiple geographically distributed data centers connected via dedicated backbone networks or the Internet. Each data center receives power supply from a nearby utility. As shown in the gray boxes, our system includes two major components: a *controller* and *pricing modules* embedded within the utilities. The controller communicates with all pricing modules to inform the current power usage of the data centers and to receive new energy prices. Moreover, the controller monitors and manages all data centers to determine if the overall power cost can be reduced via job migrations. Each pricing module receives energy shift requests from its utility and interacts with the controller to determine a new power price.

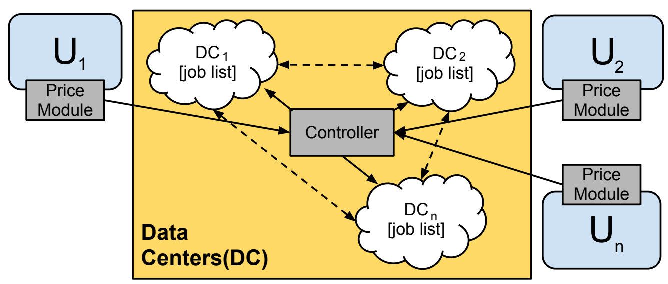


Fig. 2. Architecture

To determine concrete job migration strategies, we use Cologne [10], an optimization platform that enables *constraint optimization problems* (COPs) to be declaratively specified and incrementally executed in distributed systems. Cologne can flexibly support a wide range of policy-based optimizations in distributed systems (*e.g.*, cloud [28] and wireless networks [29]), and results in orders of magnitude less code compared to imperative implementations. Cologne uses the *Colog* declarative language, which combines distributed Datalog (used in declarative networking with language constructs for specifying goals) and constraints used in COPs. To execute *Colog* programs in a distributed setting, Cologne integrates Gecode [30], an off-the-shelf constraint solver, and the RapidNet declarative networking engine [31], [32] for communicating policy decisions among various solver nodes.

Cologne can be deployed in either a *centralized* or *distributed* mode. In the centralized mode (our proposed system currently adopts this mode), all data centers are configured by one centralized Cologne instance (our controller), which performs optimizations by taking as input the system states gathered in the network. Due to scalability issues and management requirements imposed across administrative domains, Cologne can also execute a COP in a distributed setting. In the distributed deployment mode, there would be multiple Cologne instances, typically one for each data center to communicate with its neighbors. Multiple *local* solvers coordinate with each other and each handles a partial problem to solve the global objective. Note that while we employ a centralized deployment mode here, the architecture can be easily extended to a fully distributed mode.

In the next section, we demonstrate the centralized deployment scenario and briefly discuss distributed solving.

V. System Design — The Centralized Case

We first present our formulations which optimize data center costs, followed by the pricing module for power utilities. Then we describe how the two interact with each other to achieve collaborative optimizations. Since data center workload energy consumption is at the scale of mega-Watts, without loss of generality, the energy overhead incurred by running our optimization system is negligible.

A. Optimization for Data Centers

The optimization for data centers can be formulated as a constraint optimization problem. We assume there are n data centers. Each data center i ($1 \leq i \leq n$) is issued a price P_i for power from a nearby utility. In total, m jobs are running within the n data centers, and each job j is currently located at data center D_j ($1 \leq j \leq m$). Additionally, job j is characterized by an energy consumption E_j, an input data size S_j, and a resource requirement R_j. Here, R_j could be defined as, for instance, the number of compute machines required for a typical Hadoop job. Each data center i has a maximum resource capacity of C_i. This capacity restriction prevents from migrating excessive jobs to any data center. We further consider the energy incurred due to data transfers during migration. We assume that both the source and the destination data centers would consume energy to handle a job migration, and the energy consumption rate is ER per unit of data.

Given the above inputs, the output is a job assignment A_{ji} that minimizes the overall power cost of all data centers, where $A_{ji} = 1$, if job j is assigned to data center i, and $A_{ji} = 0$ otherwise. This can be formulated as follows:

$$\textbf{minimize} \quad Cost = \sum_{1\leq i\leq n} \Big((\mathsf{RunPower}_i + \mathsf{MigPower}_i) \times P_i \Big) \tag{1}$$

$$\textbf{subject to}: \quad \forall\, 1 \leq i \leq n,$$

$$\mathsf{RunPower}_i = \sum_{1\leq j\leq m} A_{ji} \times E_j \tag{2}$$

$$\mathsf{MigPower}_i = \mathsf{MigToPower}_i + \mathsf{MigFromPower}_i \tag{3}$$

$$\mathsf{MigToPower}_i = \sum_{1\leq j\leq m,\, D_j \neq i} A_{ji} \times ER \times S_j \tag{4}$$

$$\mathsf{MigFromPower}_i = \sum_{1\leq j\leq m,\, D_j = i} (1 - A_{ji}) \times ER \times S_j \tag{5}$$

$$\sum_{1\leq j\leq m} A_{ji} \times R_j \leq C_i \tag{6}$$

$$\forall\, 1 \leq j \leq m: \sum_{1\leq i\leq n} A_{ji} = 1 \tag{7}$$

Equation (1) is the optimization objective for a data center, which seeks to minimize energy costs aggregated over all n locations. For each data center i, Equation (2) computes the power to run its jobs, $\mathsf{RunPower}_i$. Equation (3) computes the power to migrate jobs to and from a data center i, denoted by $\mathsf{MigPower}_i$, which is the sum of $\mathsf{MigToPower}_i$ (Equation (4)) and $\mathsf{MigFromPower}_i$ (Equation (5)). $\mathsf{MigToPower}_i$ is accounted for every job j whose destination data center i differs from its source location D_j (*i.e.*, $D_j \neq i$), whereas $\mathsf{MigFromPower}_i$ is for every job j that originally locates at i (*i.e.*, $D_j = i$) but the new destination differs from i (*i.e.*, not A_{ji}). Constraint (6) specifies that each data center cannot run jobs whose aggregate resource requirement exceeds its capacity. Finally, Constraint (7) ensures that each job is assigned to exactly one data center. We next present specifications of above formulations in the *Colog* declarative language.

```
goal minimize C in cost(C).
var assign(Jid,Did,V) forall toAssign(Jid,Did).

r1  toAssign(Jid,Did) <- dc(Did,Price,Capacity),
      job(Jid,Power,Size,Did1,Resource).

r2  runPower(Did,SUM<P>) <- assign(Jid,Did,V),
      job(Jid,Power,Size,Did1,Resource),
      P==V*Power.

r3a migToPower(Did, SUM<P>) <- assign(Jid,Did,V),
      job(Jid,Power,Size,Did1,Resource),
      Did != Did1,
      P==V*ER*Size.

r3b migFromPower(Did, SUM<P>) <- assign(Jid,Did,V),
      job(Jid,Power,Size,Did1,Resource),
      Did == Did1,
      P==(1-V)*ER*Size.

r4  cost(SUM<C>) <- dc(Did,Price,Capacity),
      runPower(Did,Power1),
      migToPower(Did,Power2),
      migFromPower(Did,Power3),
      C==(Power1+Power2+Power3)*Price.

r5  totalLoad(Did,SUM<L>) <- assign(Jid,Did,V),
      job(Jid,Power,Size,Did1,Resource),
      L==V*Resource.
c1  totalLoad(Did,L) -> dc(Did,Price,Capacity),
      L<=Capacity.

r6  totalAssign(Jid,SUM<V>) <- assign(Jid,Did,V).
c2  totalAssign(Jid,V) -> V==1.
```

In the above *Colog* rules, `goal` defines the optimization objective, which is minimizing the overall cost `Cost(C)`. The output of this program are variables `assign(Jid, Did, V)` that correspond to A_{ji}, meaning assigning job `Jid` to data center `Did`. The variable `V` is boolean, informing a decision if its value is 1. Similarly, rule `r2` corresponds to Equation (2), and rules `r3a` and `r3b` to Equations (4) and (5), respectively. Rule `r4` derives the optimization goal by aggregating the running and migration power obtained from rules `r2` and `r3`, respectively. Rules `r5` and `c1` together impose the capacity constraint in Equation (6). Finally, rules `r6` and `c2` ensures that each job is assigned only once, as constrained by Equation (7).

Extensions: An important advantage of *Colog* is its extensibility. The above simple model can be extended with extra rules to consider new factors. For instance, due to confidentiality (or performance requirements), some jobs may be constrained to run only within certain data centers, thus being migrated to other locations would be prohibited. We can achieve this simply by adding one *Colog* rule, below:

```
r7 trust(Jid,Did,V) -> assign(Jid,Did,V1), V1<=V.
```

In rule `r7`, `trust(Jid,Did,V)` is a table storing confidentiality information. `V` is a boolean value indicating whether job `Jid` trusts data center `Did`. The rule imposes a constraint ensuring that job assignment is possible only if the job trusts the destination data center (`V1<=V`).

Another extension is to consider job timeliness. This can be done by adding a constraint to prevent job migration if it would result in the job missing its deadline. To achieve this, we can add two attributes, *running time* and *deadline* to each job, and add a *location* attribute to each data center. Furthermore, we would measure the time delay caused by job migration via multiplying some delay factor to the distance between source and destination data centers. To better estimate the delay, we could perform traffic analysis and compute the time based on network status, which can also be expressed as *Colog* rules.

Distributed solving: Although centralized optimizations would generally output a fair solution, the downside would be its scalability (*i.e.*, long running time for large-scale problems). More importantly, sometimes a centralized controller might not even exist due to autonomy issues, if some data centers are unwilling to share confidential internal information. In such cases, a distributed solution would be necessary. As discussed in Section IV, *Colog* further enables distributed constraint solving [10], and our *Colog* specifications here are straightforward to be extended for distributed solving, which we plan to realize in our future work.

B. Pricing Module for Power Utilities

We assume there are n utilities, one supporting each data center. Each utility U_i has a power supply PS_i (sources including solar, hydro, wind, etc.), a power demand PD_i by its nearby data center, and it is currently pricing its energy at P_i. Since balancing reserves incur operational cost for utilities, having either energy surplus or shortage is undesirable. As a result, one main objective for utilities is to minimize the difference δ between power demand and supply. To achieve this, an intuitive pricing scheme is to increase the energy price under power shortage ($\delta > 0$), and decreasing it if under surplus ($\delta < 0$). To avoid frequent pricing oscillations, energy prices should stay stable within a certain time interval. Typically, the interval is at the scale of hours [27].

We propose a linear model to determine a new energy price. At utility U_i, the price for the next time interval is:

$$P_i^{'} = \alpha_i \times \delta_i + P_i = \alpha_i \times (PD_i - PS_i) + P_i \qquad (8)$$

In Equation (8), the new price $P_i^{'}$ is linear to the difference δ_i between power demand PD_i and supply PS_i. Under this model, if the current energy demand from the connected data center is more than the available supply, the price will accordingly rise by $\alpha_i \times (PD_i - PS_i)$, and vice versa. α_i ($\alpha_i \geq 0$) is a linear coefficient specific to each utility. If α is large, the new price will drastically differ, such that data centers are strongly incentivized to shift their workloads. If α is small, the price change might not be sufficient to trigger a demand shift. Being a realistic price, any negative $P_i^{'}$ will be rounded to 0. Note that for simplicity, we use the linear model between price and energy. We leave other possibilities such as quadratic and logarithm models as interesting avenues for future work.

C. System Interactions

The two aforementioned models collaborate in the following fashion. Whenever a utility requests δ power to be shifted, it exchanges information with its connected data center by performing a few rounds of interactions. Specifically, a utility first determines a new price based on the current energy price and δ under the price model given in Equation (8). Then the new price is sent to the centralized controller. Upon receiving the price, the controller computes the optimal cost via generating job migration schedules, and returns the estimated new energy consumption back to each corresponding utility. Note that during this process, the controller only estimates the energy consumption, *i.e.*, it does not yet command data centers to apply the migrations. If the new energy displacement does not meet the expectation (*i.e.*, δ), a new price will be computed, and a new round will be initiated.

There are three termination conditions for this interaction: (1) Reaching a maximum number of rounds, (2) The displacement δ is met, or (3) The price generated in round i is the same as in round $i-1$. Upon termination, the job migration schedules would be applied.

VI. Evaluation

We have implemented our proposed system and performed preliminary experiments based on the collaborative optimization models. In our evaluation, we focus primarily on the improvements (in terms of energy savings and meeting energy demands) obtained through our optimization framework. In all our experiments, we note that optimizations require low overheads: on a commodity PC, even for the largest dataset, the optimization completes within minutes and requires low memory footprint.

A. Experiment Setup

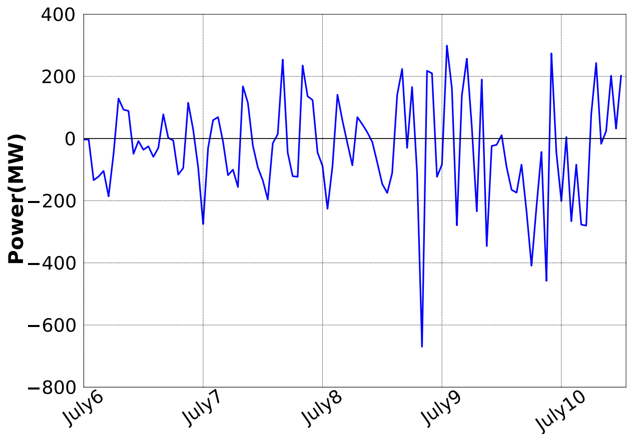
Our experimental scenario is as follows: there are four data centers owned by a single company, one of which receives power supply from Bonneville Power Administration (BPA). For the energy source, we make use of BPA's published data sets[1]. Figure 1(a) shows the errors between the predicted wind power and the actual wind power, which serve as the utility's demand shift requested to its data center. The workload within each data center is synthetically generated, consisting of 150 jobs in aggregate.

Experimental parameters. Using our model in Section V, we define our parameters as follows. Initially, the jobs are randomly assigned to these data centers. To reduce our optimization complexity, for each data center, we group jobs into batches, where each batch contains 10 jobs on average, consumes approximately 1-3 MW power and takes input of size 1-5 GB. ER (energy rate, defined in Section V-A) is set to 0.1 MW/GB and the max number of system interaction rounds

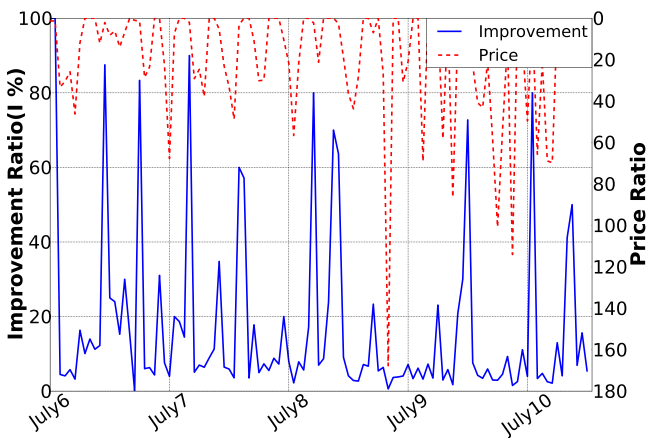
[1] http://transmission.bpa.gov/business/operations/Wind (Data Set #5)

is set to 5. We impose a practical constraint that limits each data center's energy consumption to 20 MW of energy [1]. For simplicity, we configure the resource utilization of each job to be 1 unit. Each data center has enough physical resources to accommodate at most 60 jobs. The initial price is set to 1, and α is set to 0.5 for each data center.

B. Results and Discussion



(a) Hourly Expected Energy Shift



(b) Price vs. Improvement

Fig. 3. Experiment Result

Figure 3(a) plots BPA's expected power shift on an hourly basis. In other words, it shows the target energy change that BPA hopes that data centers can adapt to dynamically. At hourly intervals, we define *Improvement I* (the percentage of final energy shift from expected energy shift) for each data center as follows:

$$I = \frac{PD_{after} - PD_{init}}{PE - PD_{init}} \times 100\% \tag{9}$$

where PD_{init} is the initial power demand of the data center. PE is the expected power from BPA (Figure 3(a)), and PD_{after} is the eventual power demand after optimization.

Figure 3(b) shows the relationship between the computed *Improvement Ratio* $I\%$ (left Y-axis in blue) and *Price Ratio* (right Y-axis in red) for a single data center, after running the optimization described in Section V. We define *Price Ratio* as a ratio of the eventual energy price (after the optimization rounds) divided by initial price. The axis has been inverted so that the correspondence between pricing and expected power shift can be easily juxtaposed.

We make the following observations from our figure. First, we observe that the expected energy shift and price have an inverse relationship. Whenever BPA has excess energy to spare (shown by a large positive energy shift value in Figure 3(a)), the price will consequently drop, in come cases to zero, indicating that data centers have an incentive to consume as much energy as possible.

On the other hand, when the expected power shift drops drastically (to -666 MW close to July 9), the price is at its highest (> 170). This triggers the data center to migrate jobs to other data centers, in order to reduce energy costs. The sharp increase in price is a consequence of using our linear pricing model.

We note that each data center has a limit on its capacity (due to BPA or other physical constraints), which consequently limits its ability to make large energy consumption shifts. For example, in our setting, each data center can consume at most 20 MW of power. Hence, the absolute improvement I is bounded by +/- 20 MW. As a result, when the target energy shift is small, we observe a larger I, resulting in a lower price. The converse is also true. When the target energy shift is large, I is smaller (due to the 20 MW bound), and that results in a higher price. The end result is an interesting complementary relationship between I and price.

Summary of Results. Overall, by averaging I at the beginning of each hour, we can observe a 17% improvement at each hour. Based on our linear pricing models, the total energy costs after the 5-day trace were 48.6 using our system, and 192.1 if data center workloads were not migrated. This results in a 3.95$\times$ cost reduction ratio. Since this is only a preliminary experiment considering the scenario for shifting energy on just one data center, by considering more data centers and further improving both models, we believe our system can achieve promising performance in the future.

VII. Conclusion

In this paper, we present the use of dynamic pricing as a mechanism for minimizing overall energy costs in data centers. Our work is motivated by Bonneville Power Administration's experiences in the Pacific Northwest, where the use of intermittent renewable energy often leads to unpredictable energy availability.

We propose a novel approach that requires collaborative optimizations across multiple data centers to balance their energy requirements, achievable through distributed constraint optimizations supported by Cologne. Our preliminary results demonstrates the viability of this approach. Moving forward, we are actively exploring how energy savings obtained from our optimizations can be realized. For instance, to reduce energy consumptions, one can reduce the number of compute units used for running cloud analytics, by prioritizing jobs based on deadlines [33]. This requires building an energy profile for each cloud application. Other techniques include migrating jobs (enqueued or currently executing) across data centers, in order to balance energy usage across different data centers.

VIII. Acknowledgments

This work is supported in part by NSF grants CNS-1117185, NSF CNS-0845552, and NSF CNS-1040672.

References

[1] Urs Hoelzle and Luiz Andre Barroso. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 1st edition, 2009.

[2] EPA Report to Congress on Server and Data Center Energy Efficiency. Technical report, U.S. EPA, 2007.

[3] Patrick Kurp. Green computing. *CACM*, 51(10):11–13, 2008.

[4] Jonathan G Koomey. Growth in data center electricity use 2005 to 2010. *Analytics Press*, 2011.

[5] Department of Energy. The Smart Grid: An Introduction. 2008. http://www.oe.energy.gov/SmartGridIntroduction.htm.

[6] S. Borenstein, M. Jaske, and A. Rosenfeld. Dynamic pricing, advanced metering, and demand response in electricity markets. *Center for the Study of Energy Markets, UC Energy Institute*, 2002.

[7] Christopher Clark Keir, Christopher Clark, Keir Fraser, Steven H, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *NSDI'05*, pages 273–286, 2005.

[8] Sherif Akoush, Ripduman Sohan, Andrew Rice, Andrew W. Moore, and Andy Hopper. Free lunch: exploiting renewable energy for computing. In *HotOS'11*, 2011.

[9] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, and Lachlan L.H. Andrew. Greening geographical load balancing. In *SIGMETRICS'11*, pages 233–244. ACM, 2011.

[10] Changbin Liu, Lu Ren, Boon Thau Loo, Yun Mao, and Prithwish Basu. Cologne: A Declarative Distributed Constraint Optimization Platform. In *Proc. VLDB Endow.*, volume 5, pages 752–763, April 2012.

[11] *Northwest Wind Integration Action Plan*. Wind Integration Forum document. Northwest Power and Conservation Council, 2007.

[12] Columbia river high-water operations. Technical report, Bonneville Power Administration, 2010.

[13] Facebook Has Spent $210 Million on Oregon Data Center, http://www.datacenterknowledge.com/archives/2012/01/30/facebook-has-spent-210-million-on-oregon-data-center.

[14] Google Data Center FAQ http://www.datacenterknowledge.com/archives/2012/05/15/google-data-center-faq.

[15] Christopher Stewart and Kai Shen. Some joules are more precious than others: Managing renewable energy in the datacenter. In *HotPower'09*. USENIX, 2009.

[16] Inigo Goiri, Ryan Beauchea, Kien Le, Thu D. Nguyen, Md. E. Haque, Jordi Guitart, Jordi Torres, and Ricardo Bianchini. Greenslot: scheduling energy consumption in green datacenters. In *Proceedings of SC'11*, New York, NY, USA, 2011. ACM.

[17] Í. Goiri, K. Le, T.D. Nguyen, J. Guitart, J. Torres, and R. Bianchini. GreenHadoop: leveraging green energy in data-processing frameworks. In *EuroSys*, 2012.

[18] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. Renewable and cooling aware workload management for sustainable data centers. In *SIGMETRICS'12*, pages 175–186, 2012.

[19] Kien Le, Ricardo Bianchini, Thu D. Nguyen, Ozlem Bilgir, and Margaret Martonosi. Capping the brown energy consumption of internet services at low cost. In *IGCC*, 2010.

[20] Changbing Chen, Bingsheng He, and Xueyan Tang. Green-aware workload scheduling in geographically distributed data centers. In *CLOUDCOM*, 2012.

[21] Navin Sharma, Sean Barker, David Irwin, and Prashant Shenoy. Blink: managing server clusters on intermittent power. In *ASPLOS '11*, pages 185–198, 2011.

[22] David Meisner, Brian T. Gold, and Thomas F. Wenisch. Powernap: eliminating server idle power. In *ASPLOS'09*, pages 205–216, 2009.

[23] Lei Rao, Xue Liu, Le Xie, and Wenyu Liu. Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *INFOCOM'10*, pages 1145–1153, 2010.

[24] HP Unveils Architecture for First Net Zero Energy Data Center, http://www.hp.com/hpinfo/newsroom/press/2012/120530c.html.

[25] David Aikema, Rob Simmonds, and Hamidreza Zareipour. Data centres in the ancillary services market. In *IGCC'12*. IEEE, 2012.

[26] David Chiu, Christopher Stewart, and Bart McManus. Electric grid balancing through low-cost workload migration. *ACM Sigmetrics Performance Evaluation Review (GreenMetrics)*, 2012.

[27] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. Cutting the electric bill for internet-scale systems. In *SIGCOMM'09*, pages 123–134, 2009.

[28] Changbin Liu, Boon Thau Loo, and Yun Mao. Declarative Automated Cloud Resource Orchestration. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, pages 26:1–26:8, 2011.

[29] Changbin Liu, Ricardo Correa, Harjot Gill, Tanveer Gill, Xiaozhou Li, Shivkumar Muthukumar, Taher Saeed, Boon Thau Loo, and Prithwish Basu. PUMA: Policy-based Unified Multi-radio Architecture for Agile Mesh Networking. In *The Fourth International Conference on COMmunication Systems and NETworkS (COMSNETS)*, pages 1–10, 2012.

[30] Gecode Constraint Development Environment. http://www.gecode.org/.

[31] Boon Thau Loo, Tyson Condie, Minos Garofalakis, David E. Gay, Joseph M. Hellerstein, Petros Maniatis, Raghu Ramakrishnan, Timothy Roscoe, and Ion Stoica. Declarative Networking. In *Communications of the ACM*, pages 87–95, 2009.

[32] RapidNet. http://netdb.cis.upenn.edu/rapidnet/.

[33] Linh T.X. Phan, Zhuoyao Zhang, Qi Zheng, Boon Thau Loo, and Insup Lee. An Empirical Analysis of Scheduling Techniques for Real-time Cloud-based Data Processing. In *4th IEEE International Workshop on Real-time Service-Oriented Architecture and Application (RTSOAA)*, 2012.