Energy-Price-Driven Query Processing in Multi-center Web Search Engines

Enver Kayaaslan
Bilkent University
Ankara, Turkey
enver@cs.bilkent.edu.tr

B. Barla Cambazoglu
Yahoo! Research
Barcelona, Spain
barla@yahoo-inc.com

Roi Blanco
Yahoo! Research
Barcelona, Spain
roi@yahoo-inc.com

Flavio P. Junqueira
Yahoo! Research
Barcelona, Spain
fpj@yahoo-inc.com

Cevdet Aykanat
Bilkent University
Ankara, Turkey
aykanat@cs.bilkent.edu.tr

ABSTRACT

Concurrently processing thousands of web queries, each with a response time under a fraction of a second, necessitates maintaining and operating massive data centers. For large-scale web search engines, this translates into high energy consumption and a huge electric bill. This work takes the challenge to reduce the electric bill of commercial web search engines operating on data centers that are geographically far apart. Based on the observation that energy prices and query workloads show high spatio-temporal variation, we propose a technique that dynamically shifts the query workload of a search engine between its data centers to reduce the electric bill. Experiments on real-life query workloads obtained from a commercial search engine show that significant financial savings can be achieved by this technique.

Categories and Subject Descriptors

H.3.3 [**Information Storage Systems**]: Information Retrieval Systems

General Terms

Algorithms, Economics, Experimentation, Performance

Keywords

Web search engine, data center, query processing, energy

1. INTRODUCTION

A major challenge in front of web search engines is to cope with the growth of the Web and the increase in user query traffic volumes while maintaining query response times under a fraction of a second. The efficiency becomes an even more critical issue as user expectations about the quality of search results and the competitive nature of the search market enforce the use of more sophisticated and costly processing techniques. Consequently, satisfying the efficiency constraints in web search necessitates the use of large compute infrastructures as well as highly efficient software platforms.

The current solution to the efficiency problem is to carry out the web search business over massive data centers, containing tens of thousands of computers [3]. Due to space and power requirements, large-scale search engines spread their infrastructures and operations across several, geographically distant data centers. The key operations in a search engine involve web crawling, indexing, and query processing. In practice, a very large index is built over the crawled web documents. Each data center maintains a replica of the most recent version of this index. Queries issued by users are processed on the index replica in the closest available data center, yielding reductions in query response latencies.

As a consequence of their massive scale, search data centers incur significant financial overheads in the forms of depreciation costs, maintenance overheads, and operational expenses, taking away a large slice of the profit made through sponsored search. Among the operational expenses, an important cost is due to high energy consumption [2]. Most standard search engine tasks (e.g., query processing, crawling, indexing, text processing, link mining) are parallelized on many computers, consuming lots of energy. This, in turn, implies high electric bills for search engine companies.

In this work, we make an early attempt to reduce the electric bills of large-scale, multi-center search engines. Our work is mainly motivated by the following two observations:

- Energy prices show high spatio-temporal variation, i.e., they differ across countries and change in time [15].
- Query workloads of search data centers show spatio-temporal variation as well, i.e., the workload of a data center varies during the day [6] and some data centers may be under high traffic while others are mostly idle.

Based on these observations, we develop a technique that shifts the query workload from search data centers with higher energy prices to those with lower prices, to reduce the total energy cost. Although the idea of shifting the workload between data centers is simple, there are two constraints that complicate the problem in the context of web search engines. First, in practice, each data center has a fixed amount of hardware resources, i.e., a data center can handle only a certain amount of query volume, at any given

SIGIR'11, July 24–28, 2011, Beijing, China.

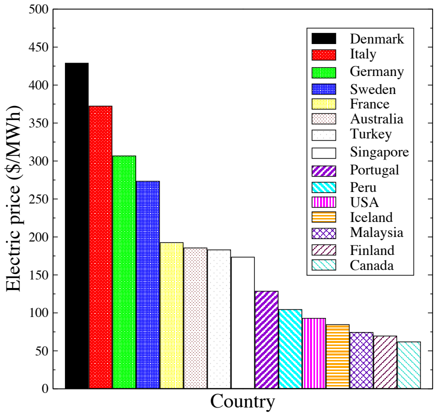

Figure 1: Electric prices in a representative set of countries in the world (source: Wikipedia).

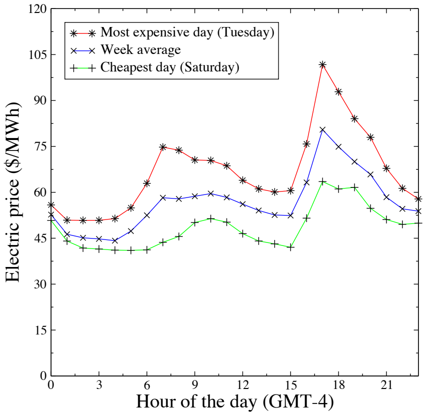

Figure 2: Hourly prices obtained over 15 different zones in the East Coast of the US (source: NYSIO).

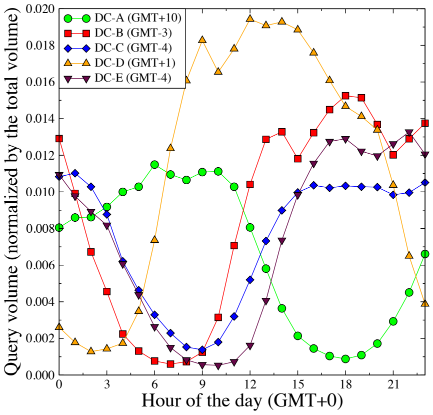

Figure 3: Hourly query traffic volumes observed on search front-ends of a commercial search engine.

time. Hence, it is not always feasible to redirect the entire search traffic volume to the data center with the cheapest electricity. Second, in web search, query response times are bounded. Hence, a query can be transferred from one data center to another only if the network latency between the data centers does not violate a query response time constraint. Our problem formulation captures both constraints.

The following summarizes our contributions. We discuss the energy-price-driven query processing problem in the context of multi-center web search engines. As a solution, we propose a probabilistic algorithm that dynamically shifts query workloads between data centers. We evaluate the proposed algorithm via simulations over a realistic search engine setting, using user queries obtained from a commercial web search engine, actual search front-ends, a large Web index, and real-life electric price data. Our results indicate significant financial savings for large-scale web search engines.

The rest of the paper is organized as follows. Section 2 motivates the idea of query workload shifting. In Section 3, we present the energy-price-driven query processing problem, together with the involved issues and performance metrics. The proposed solution is described in Section 4. We discuss, in Section 5, the details of our experimental setup. Section 6 provides the experimental results. A discussion on further issues is available in Section 7. Related literature is surveyed in Section 8. Finally, Section 9 concludes the paper.

2. MOTIVATION

Search data centers. Multi-center web search engines are known to have advantages over centralized search engines in terms of scalability and performance [7]. In practice, to further increase these benefits, the number and location of data centers should be carefully selected. The decision about data center locations is influenced by many factors, such as branding, user bases, energy prices and availability, climate, tax rates, and the political stability of countries. An important factor among these is industrial energy prices. A typical commercial web search data center consumes significant amounts of energy, which translates to an electric bill with many zeros for the search engine company.[1] Hence, there is a tendency to build data centers in countries where the energy is cheap (without completely ignoring other factors).

Variation in electric prices. Electric prices vary depending on geographical location. Fig. 1 shows the electric prices in a representative set of countries.[2] According to the figure, there is no apparent correlation between energy prices and the spatial distribution of countries. The price ratio between the most expensive (Denmark) and the cheapest (Canada) countries is about seven. Even if data center locations are restricted to the five cheapest countries, the price ratio is about 1.5. These numbers demonstrate the potential for financial savings in shifting query workloads from locations having high electric prices to cheaper locations.

Electric prices also vary in time, depending on factors such as supply/demand, capacity of transmission lines, and seasonal effects. The reader may refer to [15] for an analysis on the temporal behavior of electric prices in a wide range of markets in the US. Herein, we restrict our focus to hourly price variation within a day. As an illustrative case, we obtain the hourly electric prices from the day-ahead market of a power supplier, serving 15 zones in the East Coast of the US, and compute the hourly prices, averaged over all zones, for seven consecutive days in a week of December 2010.[3]

Fig. 2 shows the hourly electric prices for the cheapest and most expensive days, together with the average prices of the week. In general, there is high correlation across the days in terms of hourly price distribution. Prices make a peak early in the morning and late in the afternoon. Although web queries are online tasks, i.e., the processing of a user query cannot be delayed until the prices fall, temporal price variation provides further flexibility for workload shifting.

Variation in query traffic. The spatio-temporal variation in the query traffic volume provides another motivation to shift query workloads of data centers. Due to differences in time zones, some search data centers may operate under heavy workloads while others are underutilized. Moreover, the query traffic volume fluctuates throughout the day [6].

Fig. 3 shows the hourly query traffic volumes observed on five different front-ends of a commercial search engine,

[1] Some back-of-the-envelope calculations in a recent work estimate Google's annual electric bill to be $38 million [15].

[2] Wikipedia – Electricity pricing, visited on Dec. 28, 2010: `http://en.wikipedia.org/wiki/Electricity_pricing`.

[3] New York Independent System Operator, visited on Dec. 28, 2010: `http://www.nyiso.com/public/index.jsp`.

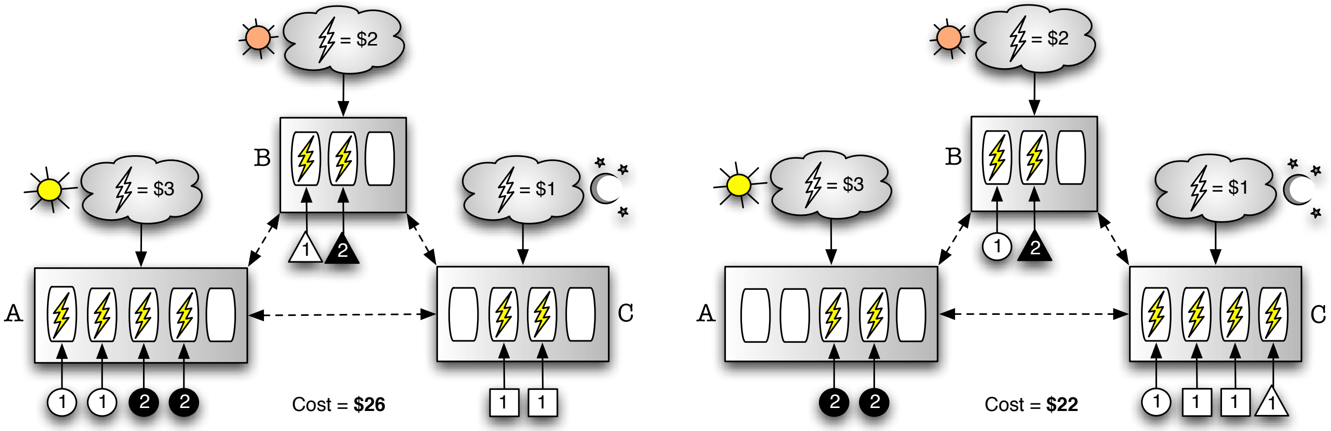

Figure 4: A sample query workload (left) is shifted between data centers to reduce the electric cost (right).

during the same 24-hour period. Query traffic volumes are observed to correlate with the local time. Typically, the volumes are higher during the day and lower during the night. As the time zone difference between two front-ends increases, the difference in their query volumes tends to increase.

Example. We informally describe our problem over the multi-center search engine setting depicted in Fig. 4. The example involves three data centers (`A`, `B`, and `C`), located in time zones that are sufficiently far apart. Data centers are connected through a wide area network. For simplicity, we assume that each data center maintains a number of homogeneous search clusters having the same compute power. In our example, data centers `A`, `B`, and `C` possess processing powers to concurrently process 5, 3, and 4 queries, respectively. In the figure, small triangles, circles, and squares represent queries issued to individual data centers at a particular instant in time. To further simplify the scenario, we assume that queries are either cheap or expensive, consuming 1 or 2 units of electricity, respectively. We also assume that expensive queries cannot be transferred between data centers due to the high overhead of network transfer, without exceeding a given hard limit on query response time. Finally, we assume that, at the instant of our example, each data center is charged a fixed cost for consuming one unit of electricity ($3, $2, and $1 for `A`, `B`, and `C`, respectively).

Given these assumptions, the initial setup on the left, where queries are processed in their local data centers, results in an electric cost of $26. By finding a better query-to-center mapping, this cost can be reduced. For example, in one extreme, if all queries are processed in `C`, which has the cheapest electricity, the total cost becomes $11. However, this is not feasible as the processing capacity of `C` can handle at most four concurrent queries. In practice, excess workload may cause intolerable delays in response time. Hence, in our work, we use data center capacities as a constraint. Moreover, we constrain response times of queries. A query is forwarded to a non-local data center only if its processing can be completed under a given time bound. In our example (on the right), two queries (one from `A` and one from `B`) are forwarded to `C`, and `A` further forwards one of its queries to `B`. However, although `B` has the capacity to process one more query, `A` cannot forward one of its expensive queries to `B`. The query-to-center mapping on the right yields the lowest possible cost ($22), without violating the constraints.

3. PROBLEM FORMULATION

The objective of the energy-price-driven query processing problem is to find a query-to-center mapping that minimizes the total electric cost incurred by a stream of web search queries. Our constraints are to keep query response times under a given time bound and to keep workloads of data centers under their capacities. Before formally specifying our problem, we introduce some notation and definitions.

Definitions. We are given a set $\mathcal{D}=\{D_1,\ldots,D_m\}$ of m data centers, a set $\mathcal{Q}=\{q_1,\ldots,q_n\}$ of n queries, and a continuous timeline T. Each data center $D_k\in\mathcal{D}$ is associated with a fixed query processing capacity C_k, which denotes the highest constant query traffic rate under which $\mathcal{D}_k$ can continue to process its queries before its waiting query queue starts infinitely growing, i.e., the capacity refers to the peak query processing throughput that can be sustained by the data center. Each query $q_i\in\mathcal{Q}$ is associated with a local data center $\widehat{D}_i\in\mathcal{D}$, a time point $t_i\in T$ at which q_i is issued by the user, its processing time c_i, and the amount of energy e_i consumed while processing the query in a data center.[4]

We are given a response time limit r that sets an upper bound on the response time of a query, i.e., r is the maximum response time that can be tolerated by users. We are also given a function $\ell_{\mathrm{uD}}:\mathcal{Q}\times\mathcal{D}\rightarrow\mathbb{R}$, where $\ell_{\mathrm{uD}}(q_i,D_k)$ denotes the network latency between data center D_k and the user who issued query q_i, and a function $\ell_{\mathrm{DD}}:\mathcal{D}\times\mathcal{D}\rightarrow\mathbb{R}$, where $\ell_{\mathrm{DD}}(D_k,D_{k'})$ denotes the network latency between data centers D_k and $D_{k'}$.[5] Moreover, we define an energy price function $\pi:\mathcal{D}\times T\rightarrow\mathbb{R}$, where $\pi(D_k,t)$ denotes the financial cost of consuming a unit of energy in data center D_k at time point t. We also define a mapping $\Phi:\mathcal{Q}\rightarrow\mathcal{D}$ that assigns each query q_i to a unique data center $D_k\in\mathcal{D}$.

We now give three definitions that are used in our problem specification. First, we define the financial cost $\psi(q_i,D_k)$ of a query q_i processed in a data center D_k (at time t) as

$$\psi(q_i,D_k)=e_i\,\pi(D_k,t). \tag{1}$$

Second, the response time $\varrho(q_i,D_k)$ for a query q_i that is eventually processed on data center D_k is estimated as

$$\varrho(q_i,D_k)=2\times(\ell_{\mathrm{uD}}(q_i,\widehat{D}_i)+\ell_{\mathrm{DD}}(\widehat{D}_i,D_k))+c_i. \tag{2}$$

[4] We assume that processing a query over the full web index consumes the same amount of energy in all data centers.

[5] Note that $\ell_{\mathrm{DD}}(D_k,D_{k'})=0$ if and only if $D_k=D_{k'}$.

Finally, the workload $\omega(D_k, t, \Phi)$ of a data center D_k (at time t) for a given query-to-center mapping Φ is defined as

$$\omega(D_k, t, \Phi) = |\{q_i : \Phi(q_i) = D_k, t \in [s_{ik}, s_{ik}+c_i]\}|, \qquad (3)$$

where s_{ik} denotes the time point that q_i is received by the data center D_k, i.e., $s_{ik} = t_i + \ell_{\mathrm{uD}}(q_i, \hat{D}_i) + \ell_{\mathrm{DD}}(\widehat{D}_i, D_k)$.

Problem definition. The energy-price-driven query processing problem is to find a query-to-center mapping Φ that minimizes the total energy cost $\chi(\Phi)$ incurred when processing a stream $\mathcal{Q}$ of queries under the mapping imposed by Φ, without violating performance constraints, i.e., to minimize

$$\chi(\Phi) = \sum_{q_i \in \mathcal{Q}} \psi(q_i, \Phi(q_i)) \qquad (4)$$

while maintaining the response time of each query q_i below the given upper-bound r on query response time, i.e.,

$$\varrho(q_i, \Phi(q_i)) \leq r, \qquad (5)$$

and enforcing that the workload of no data center D_k exceeds its capacity C_k at no time point $t \in T$, i.e.,

$$\omega(D_k, t, \Phi) \leq C_k. \qquad (6)$$

Issues. There are three issues that complicate finding a solution to our problem. First, the estimated query response time ϱ of some queries may exceed the response time limit r, i.e., it may not be possible to find a feasible solution to the problem for any given input query stream. In practice, search engines have the freedom to fully process their queries or terminate their processing early so that the response time does not exceed r. While the first approach may lead to response times not tolerable by users, the second approach may yield degraded (low quality) search results [6]. Herein, we adopt the second approach and limit the query response time to r for all queries. This way, we can always satisfy the constraint in (5) at the expense of some degraded queries.

Second, search engines do not have a control on the incoming query traffic rate, i.e., the workload of a data center is an external, uncontrollable parameter. This may prevent finding a feasible solution to our problem, as the constraint in (6) may be violated. In practice, when the user query traffic rate exceeds the peak sustainable throughput rate of the search engine, a fraction of user queries (herein, referred to as overflow queries) are dropped without any processing or processed in degraded mode, spending little time.[6] Herein, we adopt a practical scenario and assume that the overflow queries are not processed by the search engine. This way, the constraint in (6) is always satisfied at the expense of some queries that are dropped without being processed.

Third, the query stream is not available from the start. Hence, an online algorithm is required to solve the problem at hand. This implies that the decisions made by the algorithm at some point may later turn out to be suboptimal.

Metrics. We have three different performance metrics. The first metric is the objective function given in (4), i.e., the total energy cost incurred by the query mapping Φ. This is our primary metric for evaluating the quality of a solution. The second metric is the rate R_{d} of queries that are degraded to prevent the violation of the constraint in (5), i.e.,

$$R_{\mathrm{d}}(\Phi) = \frac{|\{q_i \in \mathcal{Q} : \varrho(q_i, \Phi) > r\}|}{|\mathcal{Q}|}. \qquad (7)$$

Algorithm 1 MapQueryToDataCenter(q_i, t)

```
Require: A user query $q_i$
Require: Time $t$ at which the mapping decision is made
$\mathcal{W} \leftarrow$ EstimateWorkloads$(t)$
$\mathcal{P} \leftarrow$ GenerateProbabilities$(\mathcal{W}, q_i, t)$
Select $D_k \in \mathcal{D}$ with probability $p_k \in \mathcal{P}$
$\Phi(q_i) \leftarrow D_k$
if $\varrho(q_i, \Phi(q_i)) > r$ then
    $\Phi(q_i) \leftarrow \widehat{D}_i$
end if
```

The third metric is the rate R_{o} of overflow queries that are dropped to prevent the violation of the constraint in (6), i.e.,

$$R_{\mathrm{o}}(\Phi) = \frac{|\{q_i \in \mathcal{Q} : |w'(s_{ik}, D_k, \Phi)| \geq C_k\}|}{|\mathcal{Q}|}, \qquad (8)$$

where $w'(t, D_k, \Phi)$ represents the set of queries being processed on data center D_k at time instant t, i.e., the set $\{q_i \in \mathcal{Q} : \Phi(q_i) = D_k, t \in (s_{ik}, s_{ik}+c_i], w'(s_{ik}, D_k, \Phi) < C_k\}$, where overflow queries are not included in the workload.

4. WORKLOAD SHIFTING ALGORITHM

Overview. In this section, we present an online algorithm to solve our problem, taking into account the issues mentioned in Section 3. For every given query q_i, the algorithm decides on the data center $\Phi(q_i)$ at which q_i should be processed. Steps of the algorithm are as follows. First, the query workload of each data center is estimated by the local data center, at time t. Second, estimated data center workloads are used to compute a set $\mathcal{P}$ of probabilities, where p_k denotes the probability with which q_i should be processed on D_k. Finally, the data center $\Phi(q_i)$, which will process q_i, is selected based on the discrete probability distribution implied by $\mathcal{P}$. If the estimated query response time exceeds the response time limit, the query is mapped to the local data center. Algorithm 1 provides an overview of these steps.

Estimating workloads. We assume that data centers exchange messages at regular time intervals to let others know about their current workloads. We approximate the workload of a non-local data center at a certain time by using past workload values, sampled over a period of time, from its recent workload history. In Algorithm 1, $\mathcal{W}$ denotes the set of data center workloads estimated at time t. We assume that workloads and electric prices do not significantly vary while queries are being forwarded. Hence, the workload estimated for a data center is a close approximation for the workload the data center actually has while it processes q_i.

Generating probabilities. The basic idea is to forward queries to data centers that consume cheaper electricity with higher probability, also taking the capacities and current workloads of data centers into account. We note that, in practice, it is difficult to accurately determine the actual data center workloads when deciding whether to forward a given query or not. Hence, we resort to a probabilistic approach that spreads queries across data centers to prevent workload concentration in a single data center. Given a query q_i and the set $\mathcal{W}$ of estimated data center workloads, a local data center computes (at time t) the forwarding probabilities among all data centers as follows (see Algorithm 2). Initially, for every data center D_k, the current workload L_k of D_k is set to its estimated workload W_k. The algorithm

[6] Note that forwarding of queries may lead to constraint violations as well. We will take this into account in our solution.

Algorithm 2 $\textsc{GenerateProbabilities}(\mathcal{W}, q_i, t)$

```
Require: Set $\mathcal{W}$ of estimated data center workloads
Require: A user query $q_i$
Require: Time $t$ at which the mapping decision is made
 1: for each $D_k \in \mathcal{D}$ do
 2:     $L_k \leftarrow W_k$
 3:     $p_k \leftarrow 0$
 4: end for
 5: for each $D_k \in \mathcal{D}$ in increasing order of $\pi(D_k, t)$ do
 6:     if $D_k = \widehat{D}_i$ then
 7:         $p_k \leftarrow \widehat{L}_i / \widehat{W}_i$
 8:         return $\mathcal{P} = \{p_k : D_k \in \mathcal{D}\}$
 9:     end if
10:     if $L_k < C_k$ then
11:         $\mathcal{D}' \leftarrow \{D_\ell : \pi(D_\ell, t) > \pi(D_k, t)\}$
12:         while $\mathcal{D}' \neq \emptyset$ do
13:             $\ell \leftarrow \arg\min_p \{L_p : D_p \in \mathcal{D}'\}$
14:             $s \leftarrow \min\{L_\ell, (C_k - L_k)/|\mathcal{D}'|\}$
15:             $L_k \leftarrow L_k + s$
16:             $L_\ell \leftarrow L_\ell - s$
17:             $\mathcal{D}' \leftarrow \mathcal{D}' - \{D_\ell\}$
18:             if $D_\ell = \widehat{D}_i$ then
19:                 $p_k \leftarrow s / \widehat{W}_i$
20:             end if
21:         end while
22:     end if
23: end for
```

then iterates on all data centers where the unit energy consumption cost is lower than that of the local data center $\widehat{D}_i$ (in increasing order of prices). At each iteration, the algorithm picks a data center D_k and simulates forwarding decisions from remaining data centers to D_k. First, the current workload L_k of D_k is compared with its capacity C_k to make sure that there is available capacity for additional queries. If there is unused capacity, a set $\mathcal{D}'$ of data centers whose unit energy consumption costs are higher than that of D_k is constructed. Then, until $\mathcal{D}'$ becomes empty, the data center $D_\ell \in \mathcal{D}'$ with the lowest workload is picked and removed from $\mathcal{D}'$. At each iteration on $\mathcal{D}'$, the available capacity of D_k is evenly shared among the data centers remaining in $\mathcal{D}'$. Hence, the forwarding rate s, from D_ℓ to D_k, is computed as the minimum of the current workload L_ℓ of D_ℓ and an even share of the available capacity at D_k. Subsequently, current workloads of D_k and D_ℓ are updated. If the picked data center is the local data center $\widehat{D}_i$, the probability p_k of forwarding to D_k is set to the ratio of the forwarding rate s to the estimated workload $\widehat{W}_i$. The probability that q_i is locally processed is computed as the ratio of the remaining workload $\widehat{L}_i$ of $\widehat{D}_i$ to the estimated workload $\widehat{W}_i$.

The probability generation algorithm satisfies two invariants. First, the sum of the probabilities in $\mathcal{P}$ equals to one (line 8), i.e., there is at least one data center with non-zero probability. Second, if data centers conservatively estimate the workloads of others, then no data center becomes overloaded due to forwarded queries. When simulating workloads (lines 10–22), every data center conservatively assumes that the unused capacity of a candidate center will be evenly shared among centers with higher energy prices (line 14) and only forwards as many queries as its share allows.

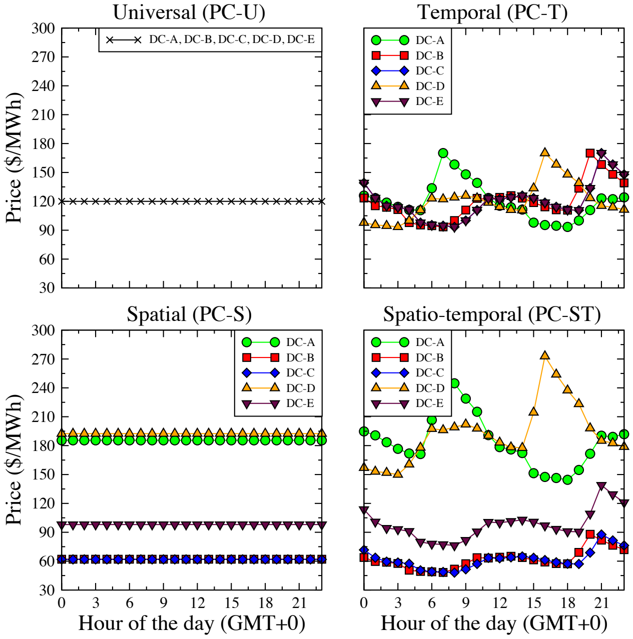

Figure 5: Electric price configurations used in the experiments: universal (`PC-U`), spatial (`PC-S`), temporal (`PC-T`), and spatio-temporal (`PC-ST`).

5. EXPERIMENTAL SETUP

Search data centers. We simulate a web search engine with five data centers, denoted as `DC-A`, `DC-B`, `DC-C`, `DC-D`, and `DC-E`. Data centers are assumed to be located in capital cities of five geographically distant countries, which are not disclosed due to the risk of revealing financially sensitive information about query traffic volumes. Network latencies between data centers as well as those between users and data centers are estimated by applying the technique described in [8]. We assume that the building blocks of data centers are identical, homogeneous search clusters. In our simulations, we determine the number of nodes in a search cluster such that each node serves about three million web documents. The number of search clusters in a data center is determined based on the query traffic volume received by the data center, i.e., each data center is associated with a separate peak sustainable query throughput value.

Result caching. We assume the presence of a query result cache with infinite capacity [6] in every data center. The result cache of a data center maintains results of queries issued by users as well as results of queries forwarded by other data centers. A time-to-live mechanism is not implemented as the performance is evaluated on a single day of queries. We assume that the time cost and energy consumption of looking up queries in the result cache are both negligible.

Electric price configurations. We generate four different price configurations (Fig. 5) based on the daily electric price distribution given in Fig. 2 and the mean of that distribution (about $120): universal (`PC-U`), temporal (`PC-T`), spatial (`PC-S`), and spatio-temporal (`PC-ST`). `PC-U` assumes that the price of electricity is fixed during the day and across all data center locations. In this configuration, all queries are locally processed since no cost saving can be achieved by workload shifting. `PC-T` assumes that prices vary during the day but not across data centers. Price distributions of data centers are identical to those in PC-U (but, they are shifted in time). PC-S assumes that prices do not vary during the day. However, there is spatial variability based on location. We obtain the spatial price variability by shifting the mean of the original price distribution by the real-life electric prices in the countries where data centers are located (using the data in Fig. 1). PC-ST is the most diverse, yet realistic configuration, where price distributions are computed by assuming both spatial and temporal price variations.

Data. We sample about 38 million queries from five different front-ends of the Yahoo! web search engine during four consecutive days (query sets S_1, S_2, S_3, and S_4). Queries in S_4 are used to evaluate our workload shifting algorithm. Those in S_2 and S_3 are used for parameter tuning purposes (see Section 6). Each set S_i is used to warm up the result cache before an experiment using the query set of day $i+1$. As the document collection, we obtain 200 million pages sampled from the Web. To prevent a mismatch between queries and the collection, we use only the documents whose predicted region matches one of the five data center locations. A proprietary region classifier is used to filter the pages.

Parameters. Each query is associated with a fixed preprocessing cost (e.g., query rewriting, spell checking), set to 20 ms. Queries are assumed to be processed over the full web index. Our algorithm is independent of the underlying ranking technique and has no impact on the search quality. The processing cost of a query is assumed to be correlated with the sum of its terms' inverted list sizes (i.e., the total number of postings) [11]. The total time needed to process a query is estimated by multiplying this cost with a per posting processing cost of 200 ns, which is an empirical value obtained from the Terrier search engine [14]. We try to keep the overflow query rate under a satisfactory value, set to 0.005 in our work (this requires the tuning described in the next section). As the query response time limit, we try several different values ($r \in \{100, 200, 400, 800, \infty\}$, in ms).

Baseline. Our baseline is the scenario in which all queries are processed in their local data centers, i.e., no query workload is shifted between data centers. We assume that a query is forwarded to non-local data center only if some reduction in the electric cost is forecasted. Ideally, some workload could have been shifted to reduce the overflow query rate even though there is no cost saving. We refrain from this kind of shifting as the primary objective of our work is to reduce the electric cost, not the overflow query rate.

6. EXPERIMENTAL RESULTS

Tuning data center capacities. In practice, data centers are given fixed compute resources, based on the query traffic volumes they receive. If the compute capacity of a data center is not carefully tuned, it may be underutilized or the overflow query rate may be too high. Hence, herein, we first perform such a tuning and assign each data center compute resources proportional with the peak query traffic volume it typically receives. When tuning, we assume that queries are not forwarded. In particular, we assign to a data center the least amount of compute resources sufficient to keep the overflow query traffic volume below a threshold.

In our setup, as shown in Fig. 6 (obtained with S_2), as data centers are given more resources, i.e., their peak sustainable throughput (PST) is increased, the overflow query rate almost linearly decreases. In our simulations, we set the PST values of a data center to the lowest value at which the overflow query rate remains below a threshold of 0.005. At this rate, the PST values we obtain are 31, 39, 34, 47, and 35 queries/sec, for the five data centers (listed in alphabetical order). In the presence of a result cache, the PST values we obtain are 13, 14, 14, 16, and 12 queries/sec. Note that lower PST values are sufficient in the latter case because a large fraction of queries are served by the cache. We use these two sets of PST values in the rest of our experiments.

Estimating data center workloads. A critical issue is to accurately estimate workloads of non-local data centers. We assume that each data center sends messages (every second) to other data centers to inform them about its current workload. The workload of a data center at a certain time is estimated by using its most recent workload history over a period of time, referred to as the window. In particular, we approximate the workload of a data center by the maximum observed workload value in its window. Note that taking the average is less conservative than using the maximum of sample workload values as it results in relatively lower workload estimates and thus higher overflow query rates. Hence, our choice of using the maximum sample value is reasonable.

In practice, the window size should be selected such that the workload estimates are as accurate as possible. Small windows may not have enough sample data while large windows may not capture the recent query traffic behavior, resulting in over-estimated workloads and hence low forwarding rates. In our work, we set the window size to the minimum possible value at which the total overflow query rate observed on the training query set (S_3) remains under the threshold we set before (i.e., 0.005). As shown in Fig. 7, when the window size is less than 10 seconds, overflow query rates are high as too many queries are forwarded to highly loaded data centers due to inaccurate workload estimates. In remaining experiments, for all data centers, we set the window size to 10 seconds for PC-T and to 16 seconds for PC-S and PC-ST. When caching is considered, we use 6, 7, and 7 seconds for PC-T, PC-S, and PC-ST, respectively.

Dissection of queries. We classify queries under degraded, overflow, non-local, and local classes. Degraded queries are those whose processing is early terminated due to the query response time limit. Overflow queries are those that are dropped as the data center does not have enough capacity. Non-local and local queries are processed in non-degraded mode at non-local or local data centers, respectively. This and remaining experiments use the test set S_4.

As seen in Fig. 8, the response time limit has a strong impact on the degraded query rate. Reasonable degraded query rates are obtained when the response time limit is larger than 400 ms. We note that the degraded query rate is independent of forwarding and is the same for all price configurations. The overflow query rate, on the other hand, is affected by these factors. However, due to the careful resource tuning mentioned before, this rate is kept under a satisfactory value in all possible scenarios (typically, 0.005, which is the permitted overflow query rate). Since the query response time limit is relaxed, there is more opportunity to forward queries between data centers. As mentioned before, no queries can be forwarded in the PC-U setup.

Impact of result caching. On aggregate, about 63% of the query traffic volume is served by the result cache. Fig. 9 shows the dissection of queries for the "miss" query traffic volume hitting the backend search systems. In general, caching reduces the degraded query rate. However, since

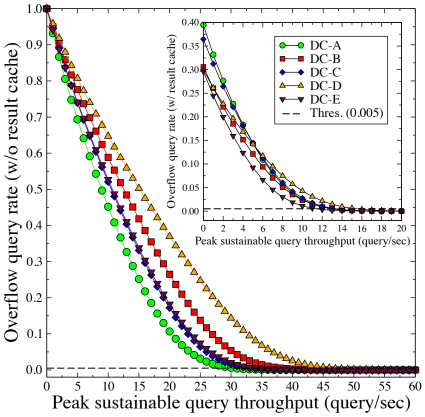

Figure 6: Overflow query rate versus peak sustainable query throughput of data centers.

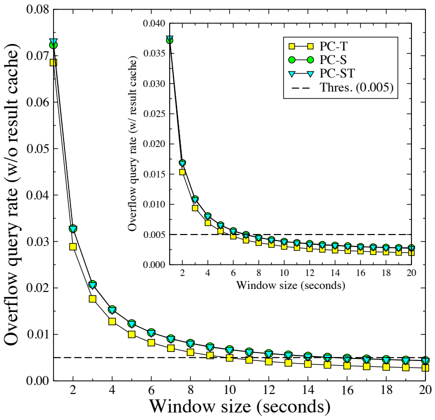

Figure 7: Overflow query rate as the window size varies.

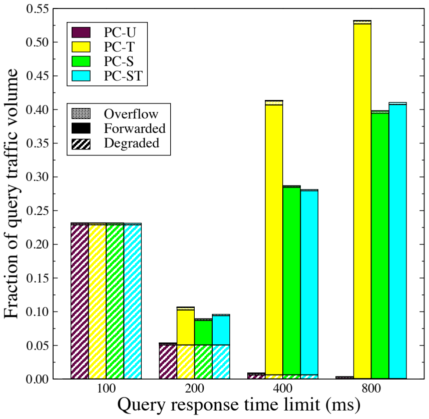

Figure 8: Degraded, forwarded, and overflow query rates (without result caching).

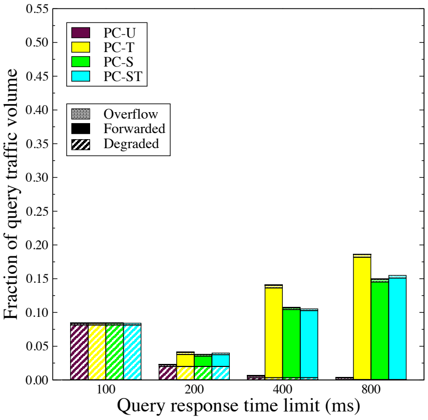

Figure 9: Degraded, forwarded, and overflow query rates (with result caching).

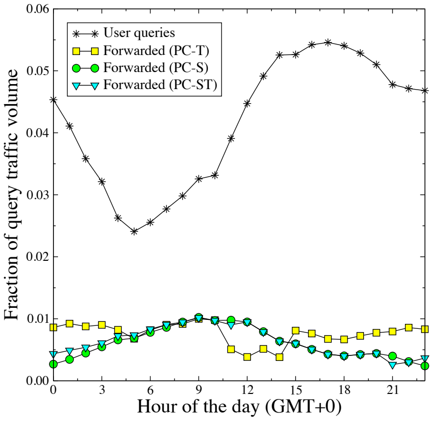

Figure 10: The rate of forwarded queries as the user query traffic volume varies ($r = 800$).

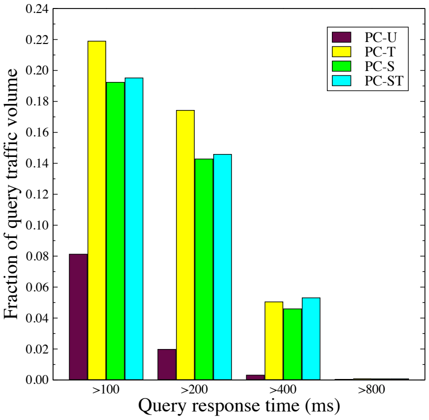

Figure 11: The rate of queries that are responded beyond a certain time delay ($r = \infty$).

data center capacities are separately tuned for the caching scenario, it has no impact on the overflow query rate, which remains around 0.005. In the rest of our experiments, we consider only the case in which a result cache is present.

Query forwarding rates. Fig. 10 shows the aggregate hourly query forwarding rate of data centers for different price setups and the variation of the hourly user query traffic volume (w.r.t. GMT+0). The reported forwarding rates are relative to the total user query traffic volume and include overflow queries, which may also be forwarded to non-local data centers. Intuitively, having more user queries implies a higher query forwarding rate. Interestingly, however, we observe that forwarding rates may drop as the user query traffic volume increases. This is because data centers have increased workloads and hence the solution space of our workload shifting algorithm is restricted. In general, the forwarding rate is more stable for the PC-T setup as price variation is lower relative to PC-S and PC-ST. We also observe that PC-ST highly correlates with PC-S, as forwarding decisions depend on the ordinal ranking of data centers according to their unit energy consumption costs, instead of the actual costs, and as electric price differences across countries are more dominant than intra-day price volatility.

Query response times. Fig. 11 shows the fraction of queries that cannot be answered under a specific amount of time, assuming there is no bound on the response time limit. According to the figure, almost all queries can be processed under 800 ms. In general, the PC-T, PC-S, and PC-ST scenarios result in higher query response times (on average, 109 ms, 103 ms, and 105 ms, respectively), relative to the average response time of the PC-U scenario (66 ms), where all queries are processed in their local data centers. Nevertheless, around only 5% of the query volume cannot be processed under 400 ms, which is a satisfactory result for the web search engine standards. This implies that, despite the large network latencies between data centers, forwarding of queries and hence reduction in electric costs is possible.

Saving in the electric cost. Fig. 12 shows our most striking result. Depending on the price setup, significant savings are achieved in electric costs relative to the respective baselines in which no workload is shifted. The largest saving (about 35% when $r = \infty$) is possible with the PC-ST scenario, which has the largest variation in electric prices.

Temporal effects. In general, the saving in the electric cost due to workload shifting is affected by the forwarding rate, which depends on the query traffic volume and electric

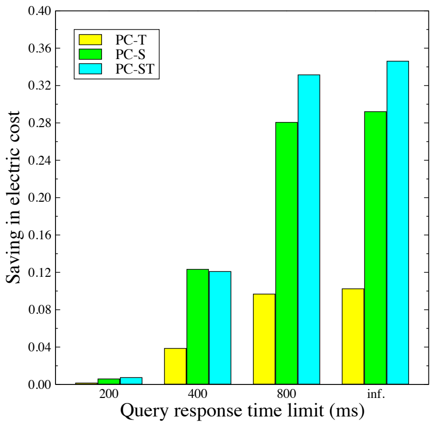

Figure 12: Saving in electric costs for different price configurations.

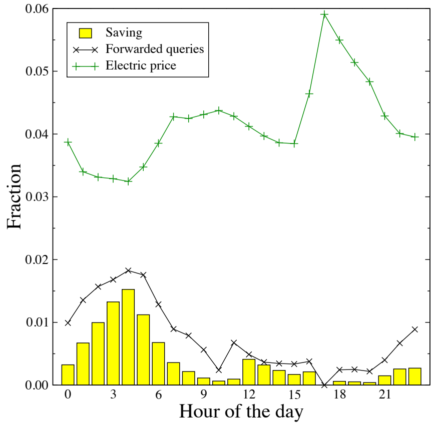

Figure 13: Temporal distribution of the cost saving (PC-T, $r=800$).

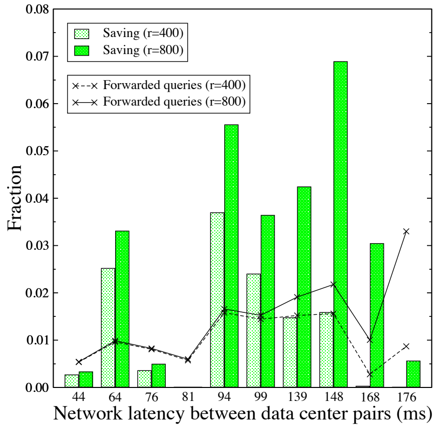

Figure 14: Spatial distribution of the cost saving (PC-S).

prices. Fig. 13 shows the distribution of the saving for the PC-T price setup, as the query forwarding rates and electric prices vary (for $r=800$ and w.r.t the local time of the data center where the query is processed). As seen in the figure, there is a high correlation between the query forwarding rate and the saving. This implies a negative correlation between the electric price and cost saving, i.e., fewer queries are forwarded when electric prices are high. It is interesting to note that, at 17:00 PM, the query forwarding rate falls to zero. At this hour of the day, the price of electricity consumed by data centers reaches the global maximum and no queries can be forwarded to a data center operating at this hour.

Spatial effects. Fig. 14 shows the distribution of the saving for the PC-S setup, as the query forwarding rate varies (the x axis shows the network latencies between data center pairs, for all possible pairs). Although not very strong, we observe some positive correlation between query forwarding rates and network latencies between data centers. This is somewhat surprising as one may expect more queries to be forwarded when the latency between two data centers is low. In practice, however, data centers with low latencies tend to be located in nearby time zones. This implies that their hourly query traffic volumes follow a similar pattern, rendering forwarding of queries more difficult. The price of electricity consumed by data centers forms another factor. As an example, consider data centers DC-D and DC-A, which consume the most expensive electricity (see Fig. 5). Although they are geographically very close (the network latency is 81 ms, in Fig. 14), these two data centers almost never forward queries to each other, as they prefer forwarding their queries to other data centers. We note that, in the PC-S setup, the entire forwarded query traffic volume between two data centers is generated by the data center that consumes cheaper energy, whereas forwarding of queries can be bidirectional in the PC-T and PC-ST setups. In general, most of the saving is obtained from data centers located in far-away time zones with a large difference in electric prices. Finally, we note that, although the network latency is not dominantly decisive in query forwarding rates, it becomes more decisive as the response time limit is reduced. In Fig. 14, we observe that, when $r=800$, a significant fraction of the cost saving is achieved due to forwarding of queries between far apart data centers. When $r=400$, however, some distant pairs of data centers are unable to exchange their workloads.

7. DISCUSSION

Impact of server utilization. The energy consumption of modern servers depends on their utilization level [4]. Shifting query workloads between data centers may have an impact on the utilization of compute servers in data centers and, in turn, affect their energy consumption. In our setup, the utilization distribution we observe when no queries are forwarded (see Fig. 15) is comparable to that reported for Google servers [5, p. 55]. However, when the workload is shifted, we obtain a quite different distribution (shown in the figure for the PC-ST scenario). We observe that data centers now have significantly more idle cycles and, in the mean time, their utilization is shifted towards higher levels.

We analyze the impact of this shift in utilization levels on the cost saving, analytically, via representative functions that map a utilization level u to a value c, indicating the energy consumption at level u relative to the consumption at peak utilization. Following [5], we evaluate functions of the form $c(u,p) = (1+u^p)/2$, where p is a free parameter ($p \in \{0.25, 0.5, 1, 2, 4\}$). Here, increasing values of p increase the energy-efficiency of compute resources running at low utilization levels. For convenience, the evaluated functions are plotted in Fig. 16.[7] The linear function $c(u,1)$ is typical for today's servers while the rest are hypothetical.

In Fig. 17, we report the respective cost savings for the above-mentioned functions. For $r = 800$, we observe that workload shifting highly benefits from increasing energy efficiency of servers. We note that the increase in idle cycles lets workload shifting benefit from power saving techniques (e.g., shutting down servers or putting them in sleep mode).

Unified cost model. Shifting workloads is not only useful for reducing the energy cost, but also for increasing data centers' availability, performance, and service quality. Ideally, all these factors should be combined under a unified cost model. In the particular case of search engines, financial implications of search result quality and search efficiency should be quantified and incorporated into this cost model.

Electric prices. We assumed that the amount of workload shifted is not significant enough to alter electric prices. However, given that important financial savings are possible,

[7] Ideally, servers would consume no energy in the absence of load. Modern servers, however, draw an idle power that is about 50% of the peak consumption [4].

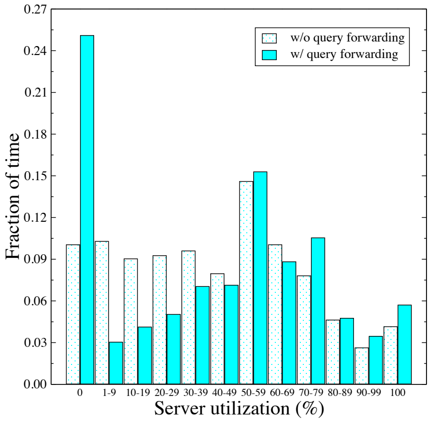

Figure 15: Fraction of time data centers run at a certain utilization level (PC-ST, $r = 800$).

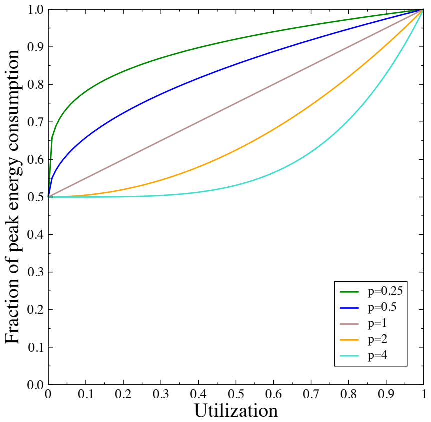

Figure 16: Mappings from utilization level (u) to energy consumption (c) (we use $c(u,p) = (1+u^p)/2$).

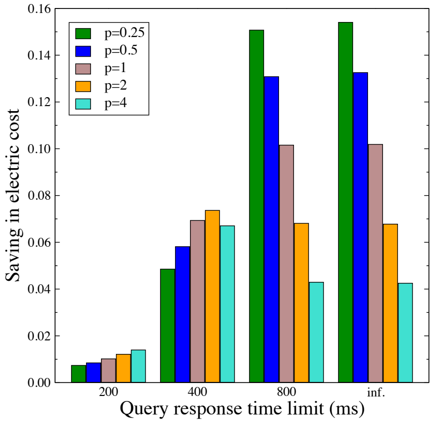

Figure 17: Saving in the electric cost for functions with varying p values (PC-ST).

many large-scale Internet services can be expected to start using similar workload shifting techniques. This may lead to an increase in prices at energy-cheap locations and vice versa, eventually converging prices to an equilibrium, which may render the technique less profitable. We also assumed a fixed electric cost model and a passive market participation strategy. A different approach is direct renegotiation of the electric price, rather than reacting to price-spotting. The reaction time to price changes has an influence on the final savings [15]. A few works [15, 17, 18] try to spot the electric price at a certain time in different electricity markets.

Environmental impact. The electric price model used by our work is rather general, allowing for different cost functions. As pointed out in [15] and [12], we may aim at reducing the environmental impact of data centers. In practice, this implies modeling the impact as a cost function (e.g., the carbon footprint of the energy the data center consumes) so that the system could decide to shift its workload to places that use renewable energy. The cost function may be time-dependent, i.e., seasonal, weekly, or even hourly.

Energy consumption. Data centers need adequate energy elasticity to achieve high cost savings when shifting their workload to data centers with cheaper energy. Some ongoing proposals to build more energy-elastic compute clusters include energy-proportional servers [4] and dynamic server provisioning techniques [13]. Another factor that might influence energy consumption is weather differentials. Cooling systems account for a significant fraction of the total energy consumption of data centers [5]. The energy consumption of chillers can be drastically reduced when the ambient temperature is low. This implies that the workload can be shifted to cooler regions that might reduce not only the energy price but also the consumption, as it is easier to cool down the heat dissipated from data centers.

8. RELATED WORK

Multi-center web search. A few works investigate the performance of multi-center web search engines [1, 7, 8]. Cambazoglu et al. [7] try to quantify performance benefits of such search engines. Baeza-Yates et al. [1] describe a multi-site search engine architecture, where the index is partitioned among data centers, also allowing partial replication of documents. Their work provides an algorithm for forwarding queries between data centers to maintain the quality of search results, relative to that of a centralized system. Cambazoglu et al. [8] propose an improved query forwarding algorithm, using linear programming. Both works have indirect consequences on reducing the energy cost, as they reduce the number of data centers involved in query processing. However, they do not directly tackle the financial aspect of the problem as they do not consider energy prices.

Workload shifting. Wang et al. [20] explore strategies to balance the load and locality in distributed systems, finding that, although algorithms that shift the workload across data centers are imperfect, using a content distribution network may provide capacity increases ranging from 60% to 90%. Along the same line, Ranjan et al. [16] show that redirecting requests to geographically distant but lightly loaded centers can reduce the response time to about a half.

Minimizing the energy cost. Wang et al. [21] try to optimize the workload, power, and cooling management of a single data center. Shah and Krishnan [19] perform an in-depth analysis of environmental and economic costs of a large-scale technology warehouse and the potential energy saving achievable when the workload is distributed across data centers. They optimize thermal workloads based on local weather, showing that the environmental burden can be reduced by up to 30%. Due to space limitations, for more related work on the topic, we refer the reader to [5]. Herein, we discuss two works that are more related to ours [12, 15].

Qureshi et al. [15] characterize the variation in electric prices and argue that data centers could exploit this for economic gains. They quantify possible gains via simulations using workloads obtained from a content provider. Our work differs from [15] in three ways. First, we provide a formal optimization framework specific to web search engines (more suitable to throughput-intensive tasks), whereas [15] provides an informal study for general-purpose Internet services (more suitable to compute-intensive tasks). Second, the algorithm in [15] is deterministic and greedy, i.e., it does not consider the issues that motivate our probabilistic approach. Third, their problem employs a bandwidth constraint, whereas ours have a query response time constraint.

Le et al. [12] propose an optimization framework for green-aware Internet services. They try to cap the brown-energy consumption via a linear-programming-based algorithm, trying to incur the least increase in costs while satisfying some service-level agreements. Our work differs from [12] in three ways. First, our optimization problem is to directly reduce the electric bill, rather than reducing financial losses since some energy caps are respected. Second, we consider the problem in a search engine setting, taking into account degraded and overflow query rates, and propose a probabilistic solution. Third, we use a detailed setup with realistic price data, real-life query workloads, real search front-ends, result caching, a large web index, and network latencies.

9. CONCLUSION

We have provided an optimization framework and a practical algorithm, based on shifting query workloads between search data centers, in order to reduce the electric bills of multi-center web search engines. We evaluated potential savings via realistic simulations. The results demonstrate that, depending on electric price distribution, electric costs of search engines can be significantly reduced by shifting their query workloads to energy-cheap data centers.

We note that, when computing the savings in electric cost, we were quite conservative in some of our assumptions. We assumed very tight capacities for data centers [10], estimated based on past query traffic volumes. In practice, data centers allow for a certain amount of slackness in their capacities (typically, about 20%), which may allow more queries to be forwarded. Moreover, we assumed that user queries are forwarded between data centers. In practice, however, search engines make use of geographically scattered request schedulers [9], which may directly identify the best data centers to contact. This may result in lower query response latencies, which implies more forwarded queries. These practical aspects should be considered as a part of future work.

Finally, we emphasize that our work has implications for many other tasks in multi-center web search engines. Primarily, electric cost optimization frameworks similar to ours should be developed for multi-site web crawling and distributed indexing tasks. In particular, it may be interesting to devise an energy-price-aware result caching framework, where invalidation predictions are made for stale cache entries based on a combination of parameters (e.g., energy cost, backend workload, age and degradedness of search results) so that the financial cost of cache misses is reduced.

10. ACKNOWLEDGMENTS

This publication is based on work performed in the framework of the Project COAST-ICT-248036, funded by the European Community. The last author of the paper is partially supported by The Scientific and Technological Research Council of Turkey under grant EEEAG-109E019.

11. REFERENCES

[1] R. Baeza-Yates, A. Gionis, F. Junqueira, V. Plachouras, and L. Telloli. On the feasibility of multi-site web search engines. In *Proc. 18th ACM Conf. Information and Knowledge Management*, pages 425–434, 2009.

[2] L. A. Barroso. The price of performance. *Queue*, 3:48–53, 2005.

[3] L. A. Barroso, J. Dean, and U. Hölzle. Web search for a planet: the Google cluster architecture. *IEEE Micro*, 23(2):22–28, 2003.

[4] L. A. Barroso and U. Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.

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

[6] B. B. Cambazoglu, F. P. Junqueira, V. Plachouras, S. Banachowski, B. Cui, S. Lim, and B. Bridge. A refreshing perspective of search engine caching. In *Proc. 19th Int'l Conf. World Wide Web*, pages 181–190, 2010.

[7] B. B. Cambazoglu, V. Plachouras, and R. Baeza-Yates. Quantifying performance and quality gains in distributed web search engines. In *Proc. 32nd Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 411–418, 2009.

[8] B. B. Cambazoglu, E. Varol, E. Kayaaslan, C. Aykanat, and R. Baeza-Yates. Query forwarding in geographically distributed search engines. In *Proc. 33rd Annual Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pages 90–97, 2010.

[9] V. Cardellini, M. Colajanni, and P. S. Yu. Dynamic load balancing on web-server systems. *IEEE Internet Comput.*, 3(3):28–39, 1999.

[10] A. Chowdhury and G. Pass. Operational requirements for scalable search systems. In *Proc. 12th Int'l Conf. Information and Knowledge Management*, pages 435–442, 2003.

[11] Q. Gan and T. Suel. Improved techniques for result caching in web search engines. In *Proc. 18th Int'l Conf. World Wide Web*, pages 431–440, 2009.

[12] K. Le, R. Bianchini, T. D. Nguyen, O. Bildir, and M. Martonosi. Capping the brown energy consumption of Internet services at low cost. In *Proc. 1st Int'l Green Computing Conf.*, pages 3–14, 2010.

[13] D. Meisner, B. T. Gold, and T. F. Wenisch. PowerNap: eliminating server idle power. In *Proc. 14th Int'l Conf. Architectural Support for Programming Languages and Operating Systems*, pages 205–216, 2009.

[14] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and D. Johnson. Terrier information retrieval platform. In *Advances in Information Retrieval*, volume 3408 of *Lect. Notes Comput. Sc.*, pages 517–519. Springer Berlin / Heidelberg, 2005.

[15] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for Internet-scale systems. In *Proc. ACM SIGCOMM 2009 Conf. Data Communication*, pages 123–134, 2009.

[16] S. Ranjan, R. Karrer, and E. Knightly. Wide area redirection of dynamic content by Internet data centers. In *Proc. 23rd Annual Joint Conf. IEEE Computer and Communications Societies*, volume 2, pages 816–826, 2004.

[17] L. Rao, X. Liu, M. Ilic, and J. Liu. MEC-IDC: joint load balancing and power control for distributed Internet data centers. In *Proc. 1st ACM/IEEE Int'l Conf. Cyber-Physical Systems*, pages 188–197, 2010.

[18] 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 *Proc. 29th Conf. Information Communications*, pages 1145–1153, 2010.

[19] A. J. Shah and N. Krishnan. Optimization of global data center thermal management workload for minimal environmental and economic burden. *IEEE Trans. Compon. Packag. Technol.*, 31(1):39–45, 2008.

[20] L. Wang, V. Pai, and L. Peterson. The effectiveness of request redirection on CDN robustness. *SIGOPS Oper. Syst. Rev.*, 36:345–360, 2002.

[21] Z. Wang, N. Tolia, and C. Bash. Opportunities and challenges to unify workload, power, and cooling management in data centers. *SIGOPS Oper. Syst. Rev.*, 44:41–46, 2010.