# Minimizing Environmental Footprints of Data Centers under Budget and Service Requirement Constraints

**KIT-MTA-2013-0002 Updated: July 29, 2013**

Waqaas Munawar, Jian-Jia Chen
Department of Informatics
Karlsruhe Institute of Technology, Germany
{munawar, jian-jia.chen}@kit.edu

Minming Li
Department of Computer Science
City University of Hong Kong, Hong Kong
minmli@cs.cityu.edu.hk

*Abstract*—**The energy consumption of data centers has been increasing, which will continue due to the increase of Internet traffic and stringent service level agreements (SLAs). Analogously, the protection of global and local environments has also driven the regulation authorities to encourage energy consumers, especially corporate entities, for the usage of green energy sources. However, the green energy is usually more expensive (up to four to five times for some cases) than the traditional energy generated from coal and petroleum. One essential problem for managing data centers, according to the greenness tendency, is to minimize the environmental penalty (or equivalently to maximize the greenness) by dispatching the requests to proper data centers under the SLA and budget constraints. This paper presents optimization techniques for dynamic workload balancing for cloud-scale data center management. We present a model for commonly found electricity tariffs for green energy and provide an efficient heuristic algorithm to maximize its usage while incorporating its intermittent availability. We evaluate the presented solution with real-life traces of electricity prices and data center workloads. Extensive evaluations support our solution's potential to minimize the environmental penalty for Internet service providers under the budget while fulfilling their SLAs.**

## I. Introduction

The awareness towards the reduction of the emission of green house gases (GHG) is increasing for the protection of global and local environments. The trend can be seen in the new legislations and recommendations by governmental agents over the world [23]. At present, the information technology (IT) sector consumes significant amount of energy. Specifically, according to a 2008 estimation, about 2% of world's GHG emissions come from this sector [33].

Consequently, government regulations are getting stricter, and private concerns (e.g., Google [10]) are taking actions on reducing their environmental footprint and striving to go greener.

This problem is more severe for data centers and cloud service providers as they need to over-design the system to the satisfy service level agreements (SLAs) during the peak loads. They have conflicting requirements of faster and more powerful but also greener processors. Moreover, for every Watt of electricity that a server uses, 1-2 Watts of electricity are typically required to cool it.

One way to control the GHG emissions is to use the greener form of energy obtained through renewable sources like wind and sun instead of coal, petroleum and nuclear. The de facto standards for such legislation have emerged to be *cap-and-trade* schemes. The essence of *cap-and-trade* schemes is that a regional 'cap' is set on the total amount of GHG emissions for all the businesses operating in the region. Within the cap, the businesses trade allowances (i.e. carbon credits) as needed. An example is Europe-wide EU-ETS [3], which is already in its third phase and aims to reduce the GHG emissions by 21% by 2020 as compared to 2005.

The limit on the total number of available credits ensures that they have a value. After each year the business must surrender enough number of credits to cover its emissions. It can keep the spare credits for its future needs or sell them to others. This flexibility ensures that emissions are cut where it costs the least to do so.

The most common credits in *cap-and-trade* schemes are

- Renewable Energy Credits (RECs): each REC represents one megawatt hour of renewable energy produced and contributed to the power grid. For example, the facilities that produce this energy can be based on wind or solar farms. The other forms of carbon credits are Certified Emission Reductions (CERs), and Voluntary/Verified Emission Reductions (VERs).
- Certified Emission Reductions (CERs): each CER is one metric ton of reduced or avoided carbon dioxide emissions. UN chose this for Kyoto Protocol [23] which has already been ratified by more than 150 nations.
- Voluntary/Verified Emission Reductions (VERs): These are the same as CERs but have less backing and certification as CERs. These often cost less than CERs but are not yet readily acceptable by regulatory authorities.

An important aspect of the *cap-and-trade* scheme is that the brown energy cap is reduced over time so that total emissions are progressively brought down. The ultimate goal is to reduce the GHG emissions to zero [3]. Keeping this goal in sight, a logical step is to start planning for zero environmental impact and many enterprises have started in this direction [11]. Hence, the optimization goal is to maximize the use of green energy.

This helps two folds. Firstly, the unused carbon credits can be sold in the open market to maximize the profit; secondly, reduced carbon footprint can be used as a marketing tool for company's image promotion. Moreover, the brown energy is not *capped* for a single enterprise, and it limits the total pollution caused by all. For individual businesses there is virtually no cap. They can buy more carbon credits to increase the limit.

Another important aspect here is that carbon credits (like RECs) are not the same as energy (measured in kWh). Both of these, i.e. energy and RECs, can be sold and bought separately. When a wind or a solar farm produces energy, it is contributed to the power grid. Such energy can then be bought like other forms of energy. The RECs produced in this process can be bought separately. The term *green energy* is actually the sum of produced energy and RECs. Hence, it costs more than brown energy due to the addition of RECs (for details, see [11]). Depending upon availabilities, the wind energy can be in the range of 6 to 16 cents per kWh. Similarly the solar energy per kWh can range from 25 cents on sunny days to 35 cents on cloudy days. In comparison, brown energy typically costs 3$\sim$4 cents per kWh [2].

Data centers, being the biggest users of electricity in the IT sector and growing rapidly [14], have a significant environmental impact. One essential problem for managing data centers, according to the greenness tendency, is to minimize the environmental penalty (or equivalently to maximize the greenness) by dispatching the requests to proper data centers under the service level agreements (SLAs) and budget constraints. There have been several results in the literature, e.g., [16], [24]–[26], [35], [36]. Most of these researches ( [24]–[26], [36]) focus on the satisfactions of the average response time. In [16], the percentile guarantees of SLAs are considered under the setting that the brown energy consumption is individually capped. In [35], the authors consider the effect of data centers' demand on market prices of electricity.

**Our Contribution:** This paper focuses on the minimization of the environmental footprint of data centers under the budget constraint and the generalized SLAs, including percentile and average response time guarantees. We present a software optimization strategy to dynamically dispatch the incoming requests from the central hub of an Internet service provider (such as Google or iTunes) to the distributed data centers. This optimization problem is multifaceted by considering many important aspects in such a setting, explained in detail in Section II. We divide the problem into subproblems to be solved individually by each data center and by the central dispatching hub. We present a practical solution that encompasses all the energy-consuming components in a data center. That includes the energy consumption from the infrastructure for networking, computation, and cooling devices. It builds on the previous work in the domain of energy conservation in data centers and is flexible enough to be applicable to data centers consisting of heterogeneous servers as well as able to accommodate different SLAs. We evaluate this with real-world workload traces from Wikipedia [30] and varying electricity prices from different regions in USA obtained from NYISO [22]. We show that this optimization problem can be effectively and efficiently solved with our greedy algorithm by relaxing the budget constraint and can be easily adopted in data centers.

## II. Background for Data Center Greenness

This section presents the background for the important aspects that have to be considered to achieve greenness in data centers.

*Varying prices of electricity and fixed energy contracts:* The prices of electricity, both green and brown, vary temporally and geographically. Moreover, the variance in energy used for the requests, i.e. the active energy component, is a significant fraction of the total energy [24]. Hence, an appropriate service placement, both temporally and spatially, can result in significant gains.

*Multiple services with different SLAs:* Data centers are expected to offer more than one service to more than one client, under different SLAs with varying degrees of QoS guarantees and with different pricing. Majority of the previous work has focused on a single data center providing a single service. The impact of multiple SLAs and multiple services being offered by a group data centers has often not been considered.

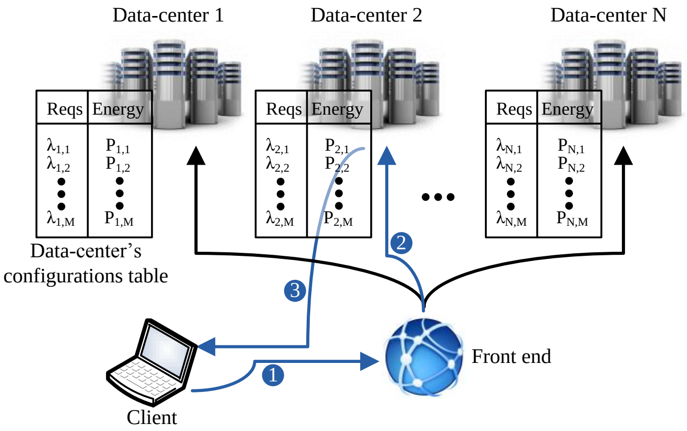
*Session-based services:* The services offered by the data centers are either session-based or stateless. In the case of session-based services, not all requests can be arbitrarily routed to any data center. The requests belonging to one session must either be served by the same data center, or the context transfer has to be considered.

*Communication latency due to geographical distance:* The distance between the data centers and the front end causes additional delay in serving the routed requests. Previous studies [24] have found that the delay is correlated to geographical distance. This delay should be considered when distributing the requests otherwise the SLA might be violated.

*Energy cost of sleep-wake transitions:* Putting a server in a data center to sleep or bringing it back for executing is not free in terms of energy consumption. Sleep-wake transitions incur additional energy costs that need to be catered when deciding to route the incoming load. By selecting a server that is already in operation, extra overhead caused by the transition can be saved.

*Energy consumption of infrastructure:* Data centers do not only consist of servers. There are also other non-computing devices as well like networking switches, routers, cooling devices and lighting. The average energy consumed by these devices is almost the same as the energy consumption of processors (typical PUE=1.9 [27]). These devices contribute substantially toward the environmental footprint of a data center and their effect must be considered.

*Energy sources and caps:* There are three basic sources of energy in each data center: (i) green energy harvested through the local resources (like a local wind farm), (ii) green energy bought in form of carbon credits and (iii) brown energy.



Data-center 1
Data-center 2
Data-center N
Reqs
Energy
$\lambda_{1,1}$ $\lambda_{1,2}$ $\lambda_{1,M}$
$P_{1,1}$ $P_{1,2}$ $P_{1,M}$
$\lambda_{2,1}$ $\lambda_{2,2}$ $\lambda_{2,M}$
$P_{2,1}$ $P_{2,2}$ $P_{2,M}$
$\lambda_{N,1}$ $\lambda_{N,2}$ $\lambda_{N,M}$
$P_{N,1}$ $P_{N,2}$ $P_{N,M}$
Data-center's configurations table
1
2
3
Front end
Client

Fig. 1: Architectural overview of a network of $N$ data centers with a typical route followed by a request and its reply

Many data centers nowadays include some local facilities to produce green energy, e.g., [4], [29]. The energy produced by the local facilities is audited and converted to carbon credits [1] which can be used just as other credits bought at local market. The price for these credits has to be paid in the form of initial expenditure on the renewable energy facility. Local wind or solar farm can produce limited supply of green energy and its maximum production cannot exceed its rated output. This can be considered as a limit on availability.

## III. Our Approach and System Model

This section first presents how the aspects related to data centers and SLAs presented in Section II are handled in the presented solution. We will then present the abstract model of the system. Then, we introduce the data center configuration table in a data center, which stores the energy consumptions in a control period for serving different amounts of workload under the SLA. We conclude this section with the energy/power sources for the data centers.

### A. Energy Contracts and SLAs

As discussed in Section II, minimizing the environmental footprint of data centers is a multifaceted problem. We present a solution that encompasses the discussed factors. The general architecture of an Internet service is presented in Figure 1, where all the data centers have individual agreements to buy energy from the power grid. These contracts can be with either varying or fixed energy prices. As fixed price energy contracts are just a special form of varying-priced energy contracts, they are both covered in our model.

Data centers offer multiple services to multiple clients under different SLAs. We can incorporate this factor by dividing the each data center into smaller cells to cover all the services that should be provided through the data centers. Each cell is considered as an individual data center. However, it is not necessary to divide the data centers to cover all the services, due to the following reasons: (1) The overhead for maintaining the coherence of the states is larger than the performance gains [17]. (2) Not all clients are geographically suitably located to be served by some of the data centers. Since the communication latency is correlated to the geographical distance [24], the clients can be statically assigned to a subset of data centers. Please note that SLAs that we consider are only within the premises of service providers. Our SLA can be combined with Internet QoS approaches to extend the guarantees all the way to the users' sites [37]. *For the rest of this paper, we only present how to deal with one SLA for the simplicity of presentation.*

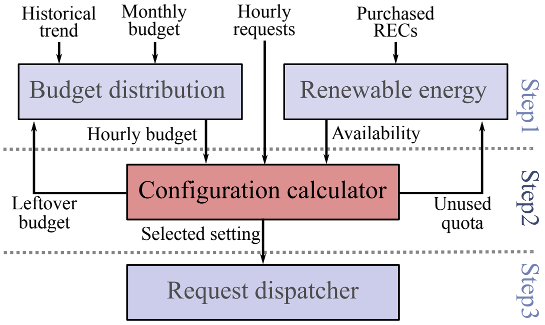


Fig. 2: Outline of proposed methodology

Incoming requests from the clients are typically distributed by a central dispatcher working on the principle of load sharing, e.g. round robin DNS [6]. We assume that once a request has been routed, the reply comes directly from the corresponding data center. If it is a session-based service, all the further correspondence is directly with the data center where the first request in the session was assigned to. In order to fulfill the SLA commitments, the internal latencies of the service must be catered for. We assume that the front end is not part of the routing process after the initial decision.

### B. System Model

We consider a network of $N$ data centers as shown in Fig 1. A central dispatcher receives all the requests and dispatches them to the $N$ data centers according to a *to-be-designed* dynamic load balancing strategy. The data centers share a common operational budget for a *budgeting period* (e.g. a month). The budgeting period is divided into smaller *control periods* (e.g. an hour). The network of data centers collaboratively provides the total required service $\lambda_{F,b}$ (the request rate) in a control period.

We tackle the problem of greening the data centers according to the methodology shown in Figure 2. Suppose that there are $B$ control periods in a budgeting period. In the first step we calculate the budget for a control period on basis of traffic forecast and green energy's availability. Predictions based on historical information or other prediction models, e.g. [5], [31], can be adopted. In the second step a load balancing strategy has to be designed for the data centers under the calculated budget constraint and the $\lambda_{F,b}$ constraint with the specified SLA. The requests are dispatched to different data centers as a result of the second step. *The main focus of our methodology in this paper is the second step, i.e. load balancing for minimizing the environmental penalty under the budget and the service requirement.*

### C. Data Center Configuration Table

Every data center requires some energy as an input to provide some service as output. The required energy consumption depends upon the service requirements as well as the hardware and infrastructure configurations of the data center. This behavior can be captured in a table for energy requirement versus the maximum service (in terms of the request rate) acceptable in a data center under the specified SLA.

We consider data centers with discretized service levels, and each level has its required energy consumption in a control period. Every data center has up to $M$ different energy usage levels (configurations) to choose from. Each energy consumption level corresponds to a particular maximum satisfiable service requirement. A data center $i$, in its $k^{th}$ configuration uses $E_{i,k}$ kWh of energy to satisfy $\lambda_{i,k}$ service requirement, under the given SLA. Once these tables have been generated for all participating data centers, the energy requirement to satisfy the contracted SLA for a given workload can be simply looked up in this table.

Possible approaches for considering the energy consumption of the servers under an SLA for a data center can be found in the literature, e.g. the methodologies in [12] or [8]. The energy consumed by infrastructure is also part of the total energy consumption $E_{i,k}$. *The data center configuration table forms the basis of a very general solution*. It can include the energy spent on cooling, the energy consumption of network equipment, the hardware heterogeneity and various settings of SLAs. It can potentially capture most of the relevant aspects of a data center with selectable granularity.

Another important aspect is the energy cost for the off→on transitions of the servers in the data centers. We assume that the entries in a data center configuration table already include the worst-case energy requirement for such transitions. Hence, we do not explicitly include them in the model. Since the transition only occurs once ($\sim$1 min [17]) per control period (1 hour in our model), i.e. turning the required servers on at the beginning of every control period, adding such worst-case energy requirements does not increase the actual energy consumption significantly.

For notational brevity, if the available energy configurations of data center $i$ is $m$ and $m < M$, we define $\lambda_{i,j} = \lambda_{i,m}$ and $E_{i,j} = E_{i,m}$ for $m < j \leq M$. Without loss of generality, with respect to $k$, we also assume that $\lambda_{i,k}$ is non-decreasing and $E_{i,k}$ is non-decreasing as well. We assume that the first entry $\lambda_{i,1}$ in the data center configuration table for data center $i$ is 0. The corresponding energy consumption $E_{i,1}$ may be 0 when the infrastructure and the hardware does not consume any energy when the data center is not used in the control period. However, practically, $E_{i,1} > 0$ and represents the energy cost of network infrastructure and other equipment, e.g. lighting, etc. In essence, it is an offset that can be added to all the entries of the configuration table.

### D. Energy/Power Sources

We consider that each data center has $Z$ different energy sources to choose from. These can be different forms of green or brown energy sources. The cost to buy a unit ($ per kWh) from the $j^{th}$ energy source in data center $i$ during control period $b$ is $C_{b,i,j}$. We assume that $C_{b,i,j}$ is time varying. Data centers with local green energy production facilities have to bear the initial investment and continous managment costs for such facilities. These costs, amortized over time, can be considered as the price of green energy.

When one unit of energy (kWh) is purchased from the $j^{th}$ energy source in data center $i$, the associate penalty is defined as $\phi_{i,j}$. In general, green energy sources have no environmental penalty associated with it, while the brown energy source has a positive penalty.

The availability of renewable energy and carbon credits in the local market depends on the local weather conditions and the cap set by the legislation authorities. Availability affects the price of energy and the cap enforces an upper limit. We assume the $j^{th}$ energy source in all data centers is limited to maximum usage of $L_j$ in the current budgeting period.

## IV. Problem Definition and Future Prediction

### A. Problem Statement

The objective is to *minimize the total environmental penalty* in the current budgeting period while satisfying the *service requirement* $\lambda_{F,b}$ (incoming request rate in every control period $b$ with $1 \leq b \leq B$) with the quality of service (QoS) as contracted in the SLA, without exceeding the *total budget* $S$ with the time varying energy prices. Each data center can choose a fraction of the total required energy in the period from any of the available sources. The optimization goal is to select an index $k_i$ with $1 \leq k_i \leq M$ for data center $i$ such that the total environmental penalty is minimized under the service requirement constraint $\sum_{i=1}^{N} \lambda_{i,k_i} \geq \lambda_{F,b}\ \forall b$ and the budget constraint.

Summarizing this, we have

| | | |
|---|---|---|
| $i, j,$<br>$k, b$ | = | Indices used for data centers, energy sources, configurations and control periods respectively |
| $N$ | = | Total number of data centers |
| $M$ | = | Max number of configurations per data center |
| $Z$ | = | Max type of energy sources |
| $B$ | = | Max control periods in a budgeting period |
| $L_j$ | = | (kWh) Maximum energy availability from $j^{th}$ source for all data centers combined |
| $S$ | = | ($) total allowed cost budget for all data centers |
| $E_{b,i,k}$ | = | Total energy requirement at data center $i$ during the $b^{th}$ control period (unit: kWh) |
| $\phi_{i,j}$ | = | (kg of $CO_2$) penalty associated with $j^{th}$ energy source in $i^{th}$ data center |
| $C_{b,i,j}$ | = | ($ per kWh) cost of $j^{th}$ energy source in $i^{th}$ data center during the $b^{th}$ control period |
| $\lambda_{F,b}$ | = | total service required during the $b^{th}$ control period |
| $\lambda_{i,k}$ | = | service provided at data center $i$'s $k$th conf. |

$x_{b,i,j}$ = In $i^{th}$ data center, portion of $j^{th}$ energy source to fulfill the energy requirement during the $b^{th}$ control period

$y_{b,i,k}$ $\in$ $\{0, 1\}$ for all $b, i, k$

With these, the problem can be stated as follows:

$$\text{Minimize: } \sum_{b=1}^{B}\sum_{i=1}^{N}\sum_{j=1}^{Z}\sum_{k=1}^{M} y_{b,i,k} \cdot E_{b,i,k} \cdot x_{b,i,j} \cdot \phi_{i,j}$$

such that,

$$0 \leq x_{b,i,j} \leq 1 \text{ for all } b, i, j \tag{1}$$

$$\sum_{j=1}^{Z} x_{b,i,j} = 1 \text{ for all } b, i \tag{2}$$

$$\sum_{b=1}^{B}\sum_{i=1}^{N} x_{b,i,j} \cdot E_{i,j} \leq L_j \text{ for all } j \tag{3}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{M} y_{b,i,k} \cdot \lambda_{i,k} \geq \lambda_{F,b} \text{ for all } b \tag{4}$$

$$\sum_{b=1}^{B}\sum_{i=1}^{N}\sum_{j=1}^{Z}\sum_{k=1}^{M} y_{b,i,k} \cdot E_{b,i,k} \cdot x_{b,i,j} \cdot C_{b,i,j} \leq S. \tag{5}$$

These can be restated as:

1) Usage of any energy source in a data center in any control period can not be more than total energy requirement for that data center in that control period.
2) Sum of all the portions from all the energy sources should satisfy the energy requirements of the data center.
3) Usage of any energy source cannot exceed its availability in the local market.
4) Total provided service should satisfy the required service for all control periods.
5) The sum of the costs occurring at the data centers should remain within the overall budget.

### B. Hardness

We now prove that the problem described in Section IV-A is $\mathcal{NP}$-hard even for deriving a feasible solution even for a single control period.

*Theorem 1:* Finding a feasible solution under a control period is $\mathcal{NP}$-hard for ensuring the feasibility conditions under the budget and service requirement constraints.

*Proof:* We reduce from the decision version of the knapsack problem. For an input instance of the knapsack problem, we are given $N$ items and two constants $W$ and $V$, in which each item $i$ has a weight $w_i$ and a value $v_i$. The objective of the knapsack problem is to select a subset of the $N$ items such that the total weight of the selected items is less than or equal to $W$ and their value is larger than or equal to $V$. The knapsack problem is $\mathcal{NP}$-complete [15].

The reduction works as follows: We construct $N$ data centers such that each data center has only two configurations for the performance and energy consumption. That is, for data center $i$, $\lambda_{i,1} = 0, E_{i,1} = 0, \lambda_{i,2} = v_i, E_{i,2} = w_i$. The performance requirement in current budgeting period $b$, $\lambda_{F,b}$ is set to $V$, while the budget is set to $W$. The cost to buy one unit from the brown energy source is set to $1$ as well.

Therefore, there exists a feasible solution for the knapsack problem if and only if the reduced instance for the studied problem has a feasible solution. As a result, we conclude that deriving a feasible solution under budget and performance constraints for the studied problem is $\mathcal{NP}$-hard. ■

Please note that the reduction in Theorem 1 relies on one assumption that the number of data centers is not a constant. Moreover, finding a solution with the minimum penalty is also $\mathcal{NP}$-hard.

### C. Infeasibility due to Unknown Future

A solution to the problem detailed in section IV-A will result in the optimal reduction in environmental penalty under the budget and performance constraints. However, to solve this problem we need $\lambda_{F,b}$ and $C_{b,i,j}$ for all future control periods. For an optimal solution these parameters must be known with certainty in advance. This is, however, not possible. Electricity prices change on hourly basis and the horizon for "certain" knowledge spans only an hour in future. Same can be argued about the incoming service requests. Service request follows long term (monthly) and short term (hourly) trends (see Figure 3). On basis of this we assume that a good enough prediction is possible only for an hour in advance. Based on these factors the problem can be transformed such that usage of green energy is maximized within a *single control period*. The modified problem is given as follows (using the modified versions of the mentioned symbols for one control period):

$$\text{Minimize: } \sum_{i=1}^{N}\sum_{j=1}^{Z}\sum_{k=1}^{M} y_{i,k} \cdot E_{i,k} \cdot x_{i,j} \cdot \phi_{i,j}$$

such that

$$0 \leq x_{i,j} \leq 1 \text{ for all } i, j \tag{6}$$

$$\sum_{j=1}^{Z} x_{i,j} = 1 \text{ for all } i \tag{7}$$

$$\sum_{i=1}^{N} x_{i,j} \cdot E_{i,j} \leq L_j \text{ for all } j \tag{8}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{M} y_{i,k} \cdot \lambda_{i,k} \geq \lambda_F \tag{9}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{Z}\sum_{k=1}^{M} y_{i,k} \cdot E_{i,k} \cdot x_{i,j} \cdot C_{i,j} \leq S. \tag{10}$$

### D. Budget distribution

The hourly budget is allocated as a weighted average of current monthly budget where the weights are calculated based on predictions. We adopt a simple prediction scheme that predicts the number of requests in the current control period based on the history as follows.

$$\lambda_n = W_n \cdot \alpha \cdot \sum_{i=1}^{B} \lambda'_i \tag{11}$$

where,

$\lambda_n$ = predicted arrival rate in $n^{th}$ control period

$\lambda'_n$ = arrival rate in $n^{th}$ control period in previous budgeting period

$B$ = Total control periods in a budgeting period

$W_n$ = Weight of $n^{th}$ control period in previous budgeting period = $\frac{\lambda'_n}{\sum_{b=1}^{B} \lambda'_b}$

$\alpha$ = correcting factor = $\frac{\sum_{b=1}^{n-1} \lambda_b}{\sum_{b=1}^{n-1} \lambda'_b}$

$\sum_{i=1}^{B} \lambda'_i$ = total traffic in previous budgeting period.

This is a quite simplistic forecast scheme and the predictions may not be very accurate. The proposed algorithm, G+D, can be used with any, more sophisticated prediction schemes, e.g., [9], [34] for better results.

In the next section we present our approach to solving this $\mathcal{NP}$-hard problem in a computationally efficient way.

## V. Our Solution

The drawback of solving the problem presented in Section IV-C separately for each control period, is that the global optimization is not guaranteed. I.e., the possibility to trade off expensive green energy in one control period against cheaper green energy in another control period might remain unutilized. We show this by solving this problem optimally within a control period through dynamic programming. After that we present a simple greedy algorithm that, by optimizing the budget distribution, produces better results in our simulations. Finally, we combine the positives of both approaches to form our final solution.

### A. Dynamic Programming (DP)

*1) Penalty Table for a Data Center:* We first consider how to optimize for data center $i$ in a control period when the local budget $S_i$ and the local service requirement $\Lambda_i$ are given. According to the definition, we know that we should choose the least power-intensive configuration of the data center that fulfills the service requirement, i.e., $k^*$ with $\lambda_{i,k^*} \geq \Lambda_i$.

Suppose that $x_{i,j}$ with $0 \leq x_{i,j} \leq \min\{1, \frac{L_j}{E_{i,k^*}}\}$ is the fraction of the total energy purchased from the $j^{th}$ energy source in data center $i$. It is now clear that the objective for this case is to minimize $E_{i,k^*} \sum\limits_{j=1}^{Z} x_{i,j} \cdot \phi_{i,j}$ such that $\sum\limits_{j=1}^{Z} x_{i,j} \cdot C_{i,j} \cdot E_{i,k^*} \leq S_i$ and $\sum\limits_{j=1}^{Z} x_{i,j} = 1$. This can be solved by using the linear programming solver in general. Since, the green energy sources have zero any environmental penalty, the above linear programming can be solved by a simple algebra calculation in $O(Z)$ time complexity given that energy sources are presorted for preference. We omit the details of algebra here.

By iterating all possible values of $S_i$ and $\Lambda_i$, we can build the corresponding penalty table $p(i, \Lambda_i, S_i)$ to show the minimum penalty for data center $i$ under the above configurations. If it is not feasible to support $\Lambda_i$ under budget $S_i$, then, $p(i, \Lambda_i, S_i)$ will be set to $\infty$.

We remove the infeasible and dominated entries in the penalty $p$-table for data center $i$ created above. An entry $p(i, \lambda, s)$ is dominated by another entry $p(i, \lambda', s')$ if $s \geq s'$, $\lambda \leq \lambda'$, and $p(i, \lambda, s) > p(i, \lambda', s')$.

Suppose that the $p$-table has $Q_i$ entries for data center $i$ after the above procedure. The $p$-table has to be generated in each control period because the penalty incurred depends on the time-varying energy prices which are not know a priori. For the $k^{th}$ entry in the $p$-table for data center $i$ with $k \leq Q_i$, we denote

- $\ell_{i,k}$ as the provided service requirement (request rates),
- $s_{i,k}$ as the allocated budget, and
- $penalty_{i,k}$ as the penalty stored in $p(i, \ell_{i,k}, s_{i,k})$.

*2) Building the Dynamic Programming Table:* On the basis of the penalty tables ($p$-table) obtained for each data center in previous step we can now build a dynamic programming table to select the appropriate configuration of every data center to provide the total required service.

Suppose that $P(i, \lambda, s)$ is the minimum penalty for the *first* $i$ data centers under the budget $s$ to provide the service requirement (total request rate) $\lambda$. For brevity, when $\lambda < 0$ or $s < 0$, we define $P(i, \lambda, s)$ as $\infty$. Clearly, for $\lambda \geq 0$ and $s \geq 0$, we know that

$$P(1, \lambda, s) = p(1, \lambda, s). \tag{12}$$

Where $p$-table is from previous section.

For $i = 2, 3, \ldots, N$, the following recursive formula can be adopted to minimize the total penalty $P$ under budget $s \geq 0$ and service requirement $\lambda \geq 0$:

$$P(i, \lambda, s) = \min_{k=1,2,\ldots,Q_i} \{P(i-1, \lambda - \ell_{i,k}, s - s_{i,k}) + penalty_{i,k}\}. \tag{13}$$

Clearly, $P(N, \lambda_F, S)$ is the minimum penalty for distributing the requests and the budgets. The standard dynamic programming technique can be adopted and the solution can be obtained via backtracking from $P(N, \lambda_F, S)$. The time complexity for calculating a single entry $P(i, \lambda, s)$ based on Equation (13) is $O(Q_i)$. To build the table correctly, we have to calculate $P(i, \lambda, s)$ from $i = 1, 2, \ldots, N$ and from $\lambda = 0$ to $\lambda_F$ and from $s = 0$ to $s = S$ sequentially. This gives the overall time complexity $O(NS\lambda_F Q_{\max})$, where $Q_{\max}$ is $\max_i Q_i$.

*Optimality and Complexity:* The above presented DP approach derives the optimal solution to minimize the environmental penalty for a control period. However, in the problem scale, some level of discretization in both budget and service is mandatory. Appropriate discretization results in a smaller global penalty table ($P$) and this reduces the computation complexity. The construction of the table $P$ depends on how we discretize the values of $\lambda$ from 0 to $\lambda_{F,b}$ and the values of $s$ from 0 to $S$. The complexity can be reduced by rounding down $s_{i,k}$ and $s$ to the nearest integer multiple of a given number, let's say, $I_s$. That is, $s'_{i,k}$ is $\left\lfloor \frac{si,k}{I_s} \right\rfloor I_s$. Similarly, we can also round down $\ell_{i,k}$ and $\lambda$ to the nearest integer multiple of a given number, let's say, $I_\lambda$. That is, $\ell'_{i,k}$ is $\left\lfloor \frac{\ell i,k}{I_\lambda} \right\rfloor I_\lambda$. Then $I_s$ and $I_\lambda$ can serve as the discretization factors of budget $S$ and $\lambda_F$. This makes the time complexity to $O(N\frac{S}{I_s}\frac{\lambda_F}{I_\lambda}Q_{\max})$.

However the price to be paid for reducing the complexity is in form of (1) over-budgeting and (2) over-provisioning. For a control period, it is not difficult to see that the over-budgeting is at most $I_sN$ and the over-provisioning is at most $I_\lambda N$. This may lead to the total over-budgeting with $I_sNB$ in the budgeting period. For data center owners budget violations can be a point of concern. However, in this solution the maximum budget violation can be reduced as much as required by finer granularity but with higher computation complexity.

Discretization does not affect the optimality of the solution derived from the DP. With the increased budget we either obtain a better result in terms of environmental penalty or the same as the optimal.

### *B. Greedy Algorithm*

We now present a heuristic algorithm based on a greedy strategy without building the penalty $p$-table constructed in Section V-A1. The two important factors to be considered are the penalty and the budget. These two factors are inversely related, i.e. to reduce penalty more budget has to be paid and vice versa. We devise a heuristic strategy which strives to minimize the weighted sum of both.

Suppose that the data center $i$ has been decided to use the $k_i^{th}$ configuration. That is, it will provide $\lambda_{i,k_i}$ service with $E_{i,k_i}$ energy consumption. Suppose that $x_{i,j}$ with $0 \leq x_{i,j} \leq \min\{1, \frac{L_j}{E_{i,k_i}}\}$ is the fraction of the total energy purchased from the $j^{th}$ energy source in data center $i$. If $k_i$ is given for every data center $i$, the objective for this case is to

$$\text{minimize} \sum_{i=1}^{N} E_{i,k_i} \sum_{j=1}^{Z} x_{i,j} \cdot \phi_{i,j} \tag{14a}$$

$$\text{such that} \sum_{i=1}^{N} \sum_{j=1}^{Z} x_{i,j} \cdot E_{i,k_i} C_{i,j} \leq S, \tag{14b}$$

$$\sum_{j=1}^{Z} x_{i,j} = 1, \qquad \text{for all } i \tag{14c}$$

$$\sum_{i=1}^{N} E_{i,k_i} \cdot x_{i,j} \leq L_j. \qquad \text{for all } j \tag{14d}$$

The above linear programming can be solved optimally by using a linear programming solver or via linear algebraic calculation with less time complexity. We omit the details for the algebra due to the space limitation.

The algorithm works as follows: all the data centers are set to their lowest service setting, i.e. $k_i = 1$ and we check for feasibility of this setting in terms of budget and service by verifying the feasibility and solving the optimal solution for Equation (14). If $\sum_{i=1}^{N} \lambda_{i,k_i}$ is no less than $\lambda_F$, the algorithm terminates; otherwise it increases one data center $i^*$ among the data centers to the next configuration $k_{i^*} + 1$. The selection of $i^*$ is as follows:

Suppose that the current solution has set $k_i$. By advancing only data center $i$ to the configuration $k_i + 1$, we can find the optimal setting in Equation (14) for minimizing the penalty under this setting. Please note that the penalty is set to $\infty$ if there is no feasible solution for Equation (14). By advancing the configuration of data center $i$, suppose that $\Delta_i^{service}$ is additional service, $\Delta_i^{penalty}$ is the additional penalty, and $\Delta_i^{budget}$ is the additional budget (this is none-zero when the budget has not yet been exhausted in the current solution).

**Algorithm 1**: The greedy algorithm

```
Input: Data center configuration table for all data centers,
       Service requirement: $\lambda_F$, Budget: $S$, weights: $w_b$, $w_e$
Output: Configuration for all data centers: $k_i$
$k_i \longleftarrow 1$ for each data center $i$;
while true do
    if $\sum_{i=1}^{N} \lambda_{i,k_i} \geq \lambda_F$ then
        if Equation (14) has a feasible solution then
            return the solution $k_i$ for each data center $i$ with
            the purchase plan by solving Equation (14)
            optimally;
        else
            return the solution $k_i$ for each data center $i$ but
            with "over budgeting" by buying all energy from
            the cheapest brown source;
    for each data center i with $k_i < M$ do
        $\Delta_i^{service} \longleftarrow \lambda_{i,k_i+1} - \lambda_{i,k_i}$;
        calculate $\Delta_i^{budget}$, $\Delta_i^{penalty}$ based on Equation (14);
    let $i^*$ be the minimum $(\frac{\Delta_{i^*}^{penalty}}{\Delta_{i^*}^{service}} \cdot w_b + \frac{\Delta_{i^*}^{budget}}{\Delta_{i^*}^{service}} \cdot w_e)$;
    $k_{i^*} \longleftarrow k_{i^*} + 1$;
```

For a data center $i$, we define two terms: *brownness*, i.e. penalty caused per unit of provided service $(\frac{\Delta_i^{penalty}}{\Delta_i^{service}})$ and *economy*, i.e. budget spent per unit of provided service $(\frac{\Delta_i^{budget}}{\Delta_i^{service}})$. The heuristic that we use is $brownness \cdot w_b + economy \cdot w_e$. Where $w_b$ and $w_e$ are the weights that can be assigned to prefer brownness over economy or vice versa.

Algorithm 1 presents the pseudo-code of the above greedy algorithm. The worst-case number of combinations that we have to check for different $k_i$ in this algorithm is $O(N^2M)$, as in each while loop in Algorithm 1 we consider up to $N$ data centers and the number of iterations in the while loop is at most $NM$. For each combination, we have to solve Equation (14). This can be sped up by starting based on the current solution. However, solving Equation (14) by using linear programming solvers is already quite efficient. As we are not able to guarantee the budget satisfaction, over budgeting may be needed, as also presented in the pseudo-code. For over budgeting the budget is borrowed from the future invocations of the same budgeting period.

### *C. Greedy and DP Combined (G+D)*

The greedy algorithm presented earlier, when allowed over-budgeting, guarantees to find a feasible solution, if there exists one. It keeps increasing the offered service progressively in search of a feasible solution. In the worst case, it configures all the data centers to run at maximum service setting. However, in the average case, it finds a feasible setting much earlier. Moreover, the heuristic used for the greedy algorithm does not buy overly expensive green energy, resulting in a efficient budget usage. In comparison, the DP method finds the optimal solution in terms of environmental penalty, even if the cost to reduce the environmental penalty is overly prohibitive.

We devise a method to combine both approaches to accumulate the benefits of both, as follows. For a given control period we execute the greedy algorithm to find a feasible solution. We analyze the budget requirement of this solution and set this as the maximum budget constraint for the DP method. Since the greedy algorithm optimizes for the budget as well, its solutions are more miserly in terms of budget usage. Setting this budget as upper limit for DP results in a reduced search space for dynamic programming approach. In this way we achieve a solution which incorporates the budget optimization of the greedy algorithm with the optimal search for minimal environmental penalty from DP approach.

As G+D uses greedy and DP sequentially, its worst case time complexity is $O(N^3 M \frac{S}{I_s} \frac{\lambda_F}{I_\lambda} Q_{max})$, using the previously introduced symbols.

In the following sections we present our evaluation results.

## VI. Simulation Setup

We adopt the settings from [36] to evaluate the proposed solution by simulating the Google's setup for the location of data centers in the US. Based on these locations, we obtain the electricity pricing information from [22]. For our simulations, the following factors play important roles.

**Non-varying factors** include the hardware capabilities of the data centers. These include server capabilities and cooling infrastructure. We consider four data centers, in which each data center is equipped with homogeneous servers, as detailed in Table I. We use the method in [32] to build the data center configuration table, presented in Section III-C, by considering 50 servers in each data center. The resulting table has at most 87 entries in each data center. Other methodologies like [12] and [8] can also be adopted for calculating the data center configuration tables. Please note that the complexity of the presented solutions does not directly depend on the number of servers in data centers, but the number of entries in the data center configuration tables. Even when the number of servers in a data center increases, we can also reduce the number of entries in the data center configuration tables by changing the granularity for management.

The penalty for a green energy source is set to $0$. The penalty for a brown energy source is set to $1$. This multiplied by $CO_2$ kg generated per kWh gives actual penalty.

**Time-varying factors** include brown and green energy prices. The availability for both the forms of energy does not vary. The fluctuation in the production of green energy due to environmental factors causes a shift in its price but the overall availability contracted by the suppliers in the form of RECs and VECs is fulfilled. Green energy has a higher price than brown energy due to intermittent production and expensive facility management. In our simulation we assume a surcharge of 1.5 cents and 18.0 cents per kWh for wind and solar energy [2] respectively in addition to the brown energy price. For price trace of electricity, we use the data from NYISO [22]. Specifically, we use Day-Ahead price data for the month of November 2007 for all four regions previously mentioned.

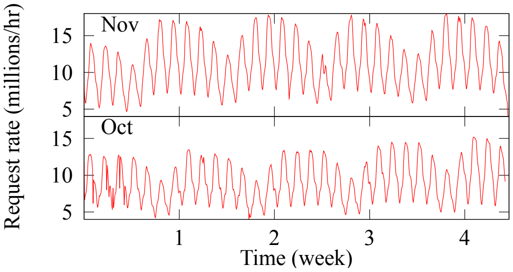


Fig. 3: Wikipedia workload trace in Oct. and Nov. 2007

The other time varying factor is the total service requirement, ($\lambda_{F,b}$). $\lambda_{F,b}$ is a random variable but overall it follows a weekly recurring pattern (see Figure 3). We use the actual workload trace from Wikipedia [30]. We use the month of October 2007 for forecasting and the month of November for the actual request trace.

## VII. Evaluation

The evaluation of the presented solution is based on the simulation setup described in Section VI. We take a month as a budgeting period and an hour as a control period. For the greedy algorithm proposed in Section V-B, we configure the heuristic weights as $w_b = 10$ and $w_e = 1$ in Algorithm 1. The presented algorithm (G+D) is evaluated for three main criteria, i.e. budget allocation and usage, environmental penalty minimization and computation time. We compare it with base line schemes of "All Green" and "All Brown" as well as dynamic programming approach (SectionV-A) and simple greedy (SectionV-B).

### A. Budget allocation and Environmental Penalty

The budget usage comparison is presented in Fig4(a). The maximum allowed budget was set to USD 80k. As expected "All Brown" uses the minimum amount of budget at the cost of huge environmental penalty, whereas, "All Green" approach violates maximum budget constraint (see Figure 4(b)).

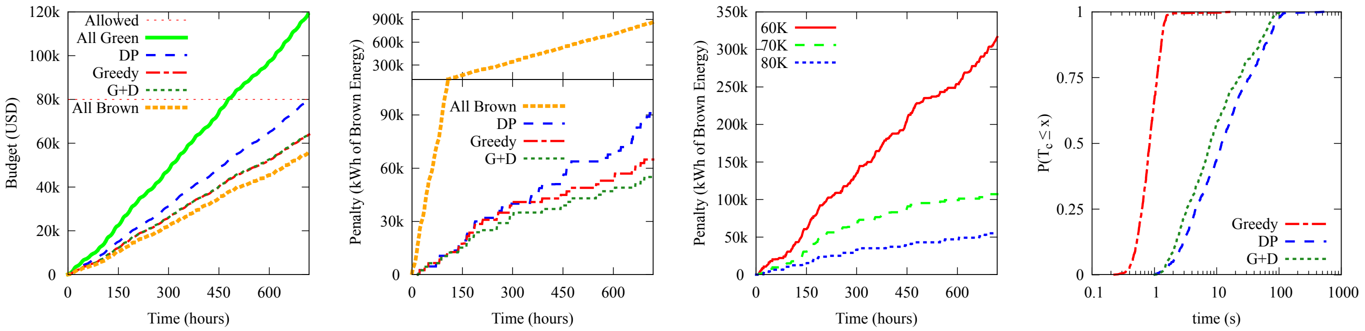
The proposed solution, G+D, follows the same budget allocation as the greedy algorithm. In comparison with the optimal budget allocating scheme, "All Brown", it uses only one eighth more budget but produces a 15 fold reduction in environmental penalty as shown in Figs. 4(a) and 4(b). In comparison with "All Green", G+D uses only half the budget.

However, the greedy algorithm uses a weighted average cost function that includes both the factors, i.e. *brownness* and *economy*. It only prefers the green energy if it is not overly expensive as discussed in Section V-B.

Relaxing the budget constraint results in decreased environmental penalty for the presented solution. The result is shown in Figure 4(c). The effect is, however, non-linear. This is

| | Data center # 1 | | | Data center # 2 | | | Data center # 3 | | | Data center # 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Location | San Luis Valley, Colorado | | | Los Angeles, California | | | Oak Ridge, Tennesse | | | Lanai, Hawaii | | |
| Processor | AMD Athlon | | | Pentium 4, 630 | | | Pentium D950 | | | AMD Athlon | | |
| Max freq. | 3.0 GHz | | | 3.0 GHz | | | 3.4 GHz | | | 3.0 GHz | | |
| | Speed ratio | Service (req/sec) | Power (W) | Speed ratio | Service (req/sec) | Power (W) | Speed ratio | Service (req/sec) | Power (W) | Speed ratio | Service (req/sec) | Power (W) |
| Power settings | 1.00 | 750 | 174.09 | 1.00 | 750 | 93.99 | 1.0 | 850 | 194.00 | 1.00 | 750 | 174.09 |
| | 0.90 | 675 | 141.28 | 0.80 | 600 | 62.76 | 0.85 | 725 | 146.19 | 0.90 | 675 | 141.28 |
| | 0.66 | 500 | 88.88 | 0.50 | 375 | 37.99 | 0.64 | 550 | 102.13 | 0.66 | 500 | 88.88 |
| | 0.50 | 375 | 68.13 | 0.40 | 300 | 34.10 | 0.44 | 375 | 78.82 | 0.50 | 375 | 68.13 |
| | 0.26 | 200 | 55.29 | 0.30 | 250 | 32.37 | 0.29 | 250 | 71.20 | 0.26 | 200 | 55.29 |

TABLE I: Data center settings used for simulation (adopted from [19]): Speed ratio is the ratio of the frequency by adopting dynamic voltage frequency scaling (DVFS) to the maximum frequency in the system.



(a) Budget usage in the month

(b) Method comparison

(c) Budget vs. penalty for G+D

(d) Computation time

Fig. 4: Evaluation Results

because increasing the monthly budget beyond a certain point makes the availability of green energy the limiting factor.

For minimizing the environmental penalty, among the presented schemes, G+D outperforms all others that follow the budgetry constraints as shown in Figure 4(b). It results in 15 fold reduction in environmental penalty in comparison with the "All Brown" baseline with only one-eighth increase in budget usage.

The dynamic programming approach uses up all the budget as soon as possible, leaving it constrained at later control periods. It is worth noting, that optimality of the dynamic programming solution is w.r.t the environmental penalty within a single control period instead of budget minimization.

The fundamental difference between G+D and the dynamic programming approach is the allocation of budget. Unlike dynamic programming, G+D tries to minimize the budget usage. This provides G+D a relaxed budget constraint progressively at subsequent control periods, as compared to the dynamic programming approach. dynamic programming produces the optimal results in terms of environmental penalty within a single control period. To this end, it sometime uses excessive budget for gaining a marginal reduction in penalty. This makes the budget constraint tighter in subsequent control periods, resulting in higher overall penalty for dynamic programming.

### B. Computation Time

For the results to be useful, the maximum computation time must remain a negligible fraction of the length of the control period. This condition can be fulfilled by lengthening the control period. But, this, in turn, makes the prediction horizon longer for $\lambda_{F,b}$ and $C_{b,i,j}$, resulting in deteriorated prediction quality and hence affects the solution quality.

One way to decrease computation time can be offline computation. But, due to the time-varying factors like price and availability of energies, offline computation is not a feasible option as it would require a huge amount of disk space to store the computation tables for (i) each possible price for each source of energy and (ii) for all the possible availabilities ($L_j$). Such tables are practically unmanageable.

Online computation of solution at the beginning of every control period is the only viable option. Figure 4(d) presents the cumulative distribution (in probability) of the computation times in one control period on a normal desktop machine (Intel i3, 8GB RAM, Linux). It is clear that the greedy algorithm is the fastest with majority of the computation times remaining within a second. However, G+D may take up to a minute in a few cases. This remains suitable for a control period of around an hour as necessitated by the electricity price horizon. Moreover, the computation time of G+D can also be reduced by changing the granularity of discretization in the dynamic programming. Furthermore, G+D is often faster than dynamic programming alone, although it uses dynamic programming as a subroutine. The is because of the reduced search space for dynamic programming that greedy algorithm prunes out.

## VIII. Related Work

Data centers being major electricity consumers in the IT sector, have been focus of lot of research to make them environmental friendly. We divide the relevant research into three main categories, as follows:

**Energy conservation:** These studies aim to decrease the energy consumption of a data center, whereas decreased environmental footprint is a side product. Examples include [7], [13], [32]. Mostly these aim to optimize a a single data center. For example Wang et al. [32] present a scheme to reduce power consumption while fulfilling the generalized SLAs within a single data center. The solution we present builds on top of these schemes as we aim for multiple data center optimization and single data center optimization is part of that.

**Electricity cost management:** These studies are more nearer to our approach. The key difference between this category and the previous one is that, here, multiple and geographically distributed data centers are considered. Examples in this category include [19]–[21], [24]. Qureshi et al. [24] were the first to tackle the problem of cost minimization by exploiting the geographic variance of energy prices but they do not consider the carbon market dynamics. These are also not considered in [19] and [20].

**Utilizing the green energy:** This is a relatively new direction with only few initial studies e.g [25], [26], [36]. Our approach falls in this category. Zhang et al. [36] present how to maximize the use of environmental friendly green energy to power the servers in data centers, while maintaining the average response time for incoming requests. However, since they use the queuing theory to model the service provision, it can not handle generalized SLAs, for instance, in the form of percentile guarantees. The same argument also applies to the limitations of the researches in [18], [25], [26]. Moreover, Rao et al. [25], [26] do not consider time-varying workloads, multiple services, or market interactions. Stewart and Shen [28] also focus on minimizing the environmental penalty by reducing the use of brown energy. They use a model in which Internet service providers own the renewable energy farm. *In contrast, we consider the more general case where the renewable energy can be locally produced or bought in form of RECs by the commercial producers and contributed to the grid.* Le et al. [16] is more thorough in their approach toward the problem. They focus on cost reduction by exploiting the distributed nature of data centers for dynamic request dispatching while maintaining SLAs. They are the first ones to consider carbon interactions. Our approach has two main differences from [16]: Firstly, we aim to maximize the green energy usage within budgetary constraints as opposed to maximizing profits within brown energy cap. Secondly, in our solution, we divide the optimization problem to smaller parts: one to be solved by each data center and the other for the front end. This helps two folds (i) we can include more factors to model energy consumption, including the infrastructure for networking, computation, cooling devices, etc., and (ii) the optimization problem can be solved more frequently because of the reduced complexity at the front end. The latter also results in a shorter horizon for energy price and traffic predictions.

## IX. Conclusion

The environmental footprint of data centers is becoming significant. In this paper we formalized the problem of minimizing the environmental footprint of ISPs (or maximizing the green energy usage) while fulfilling the budgetary and service constraints. We showed that this problem is a $\mathcal{NP}$-hard problem and presented a viable greedy heuristic for optimization. The solution that we presented (1) is up to date, in that, it is based on current legislative and economic trends. (2) It is practical. By dividing the problem into two subproblems and solving them separately, it gives us the flexibility to add different kinds of SLAs and is also valid for heterogeneous servers in a single data center. (3) It is wholistic in nature as it is cognizant of the energy usage for computation hardware, the networking hardware and also the cooling infrastructure of the data center.

We evaluated the presented solutions with traces of electricity prices and typical Internet workloads. Extensive evaluations based on real-life electricity price and Internet traffic data for multiple locations demonstrate the efficiency and efficacy of our approach.

## References

[1] North carolina renewable energy tracking system (NC-RETS). http://www.ncrets.org/.

[2] Cost-Competitiveness | solarbuzz. http://solarbuzz.com/facts-and-figures/markets-growth/cost-competitiveness, Mar 2012.

[3] The EU emissions trading system (EU ETS) - policies - climate action - european commission. http://ec.europa.eu/clima/policies/ets/index_en.htm, January 2013.

[4] Apple Inc. Apple facilities environmental report. http://images.apple.com/environment/reports/docs/Apple_Facilities_Report_2012.pdf, 2012.

[5] J. Box and G.M. Jenkins. *Reinsel. Time Series Analysis, Forecasting and Control*. Prentice Hall, Englewood Cliffs, NJ, 1994.

[6] Valeria Cardellini, Michele Colajanni, and Philip S Yu. Dynamic load balancing on web-server systems. *Internet Computing, IEEE*, 3(3):28–39, 1999.

[7] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, and R.P. Doyle. Managing energy and server resources in hosting centers. In *ACM SIGOPS Operating Systems Review*, volume 35, pages 103–116, 2001.

[8] Jian-Jia Chen, Kai Huang, and Lothar Thiele. Power management schemes for heterogeneous clusters under quality of service requirements. In *SAC*, pages 546–553, 2011.

[9] Paulo Cortez, Miguel Rio, Miguel Rocha, and Pedro Sousa. Internet traffic forecasting using neural networks. In *IJCNN'06*, pages 2635–2642. IEEE, 2006.

[10] Gary Demasi. Official google blog: Google buys next 20 years worth of wind power. http://googleblog.blogspot.com/2011/04/oklahoma-where-wind-comes-sweepin-down.html, Apr 21 2011.

[11] Google. Google's Green PPAs: What, How, and Why - Rev 02. *Google White Papers*, 29 April 2011.

[12] Raphael Guerra, Julius Leite, and Gerhard Fohler. Attaining soft real-time constraint and energy-efficiency in web servers. In *SAC*, pages 2085–2089, 2008.

[13] J. Heo, D. Henriksson, X. Liu, and T. Abdelzaher. Integrating adaptive components: An emerging challenge in performance-adaptive systems and a server farm case-study. In *RTSS*, pages 227–238, 2007.

[14] Intel CEO Paul Ontellini. Speaking at Dell World. November 2011.

[15] D.S. Johnson and M.R. Garey. Computers and intractability: A guide to the theory of np-completeness. *Freeman&Co, San Francisco*, 1979.

[16] K. Le, R. Bianchini, T.D. Nguyen, O. Bilgir, and M. Martonosi. Capping the brown energy consumption of internet services at low cost. In *Green Computing Conference, 2010 International*, pages 3–14. IEEE, 2010.

[17] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T.D. Nguyen. Managing the cost, energy consumption, and carbon footprint of internet services. In *ACM SIGMETRICS Performance Evaluation Review*, volume 38, pages 357–358, 2010.

[18] Kien Le, Ricardo Bianchini, Margaret Martonosi, and T Nguyen. Cost- and energy-aware load distribution across data centers. *HotPower*, 2009.

[19] Jie Li, Zuyi Li, Kui Ren, and Xue Liu. Towards optimal electric demand management for internet data centers. *IEEE Transactions on Smart Grid*, 3(1):183 –192, march 2012.

[20] Jianying Luo et al. Data center energy cost minimization: A spatio-temporal scheduling approach. In *INFOCOM*, pages 340–344, 2013.

[21] V. Mathew, R.K. Sitaraman, and P. Shenoy. Energy-aware load balancing in content delivery networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 954 –962, march 2012.

[22] New York Independent System Operator, NYISO. http://www.nyiso.com/.

[23] Koyoto Protocol. United nations framework convention on climate change. *Kyoto Protocol, Kyoto*, 1997.

[24] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 123–134, 2009.

[25] 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 *INFOCOM*, pages 1–9. IEEE, 2010.

[26] Amip J Shah and Nikhil Krishnan. Optimization of global data center thermal management workload for minimal environmental and economic burden. *Components and Packaging Technologies, IEEE Transactions on*, 31(1):39–45, 2008.

[27] Matt Stansberry and Julian Kudritzki. Data center industry survey. Technical report, Uptime Institute, 2012.

[28] C. Stewart and K. Shen. Some joules are more precious than others: Managing renewable energy in the datacenter. In *Proceedings of the Workshop on Power Aware Computing and Systems*, 2009.

[29] Sandra Upson. The greening of google - IEEE spectrum. http://spectrum.ieee.org/energy/environment/the-greening-of-google, October 2007.

[30] Guido Urdaneta, Guillaume Pierre, and Maarten van Steen. Wikipedia workload analysis for decentralized hosting. *Elsevier Comp. Networks*, 53(11):1830–1845, July 2009. http://www.globule.org/publi/WWADH_comnet2009.html.

[31] A. Verma, P. De, V. Mann, T. Nayak, A. Purohit, G. Dasgupta, and R. Kothari. Brownmap: Enforcing power budget in shared data centers. *ACM Middleware*, pages 42–63, 2010.

[32] Shengquan Wang, Waqaas Munawar, Jun Liu, Jian-Jia Chen, and Xue Liu. Power-saving design for server farms with response time percentile guarantees. In *RTAS*, pages 273–284, 2012.

[33] Molly Webb et al. Smart 2020: Enabling the low carbon economy in the information age. *The Climate Group. London*, 1(1):1–1, 2008.

[34] Chun You and Kavitha Chandra. Time series models for internet data traffic. In *LCN'99.*, pages 164–171. IEEE, 1999.

[35] Yanwei Zhang, Yefu Wang, and Xiaorui Wang. Electricity bill capping for cloud-scale data centers that impact the power markets. In *ICPP 2012*.

[36] Yanwei Zhang, Yefu Wang, and Xiaorui Wang. Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In *Middleware 2011*, volume 7049 of *LNCS*, pages 143–164, 2011.

[37] Weibin Zhao, David Olshefski, and Henning G Schulzrinne. Internet quality of service: An overview. 2000.