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

Waqaas Munawar[1], Jian-Jia Chen[1] and Minming Li[2]
[1]*Karlsruhe Institute of Technology, Karlsruhe, Germany*
[2]*City University of Hong Kong, Kowloon, Hong Kong*

Keywords: Green Energy Maximization, Distributed Data Centers, Response Time, Service Level Agreement.

Abstract: The energy consumption of data centers (DCs) 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 DCs, according to the greenness tendency, is to minimize the environmental penalty (or equivalently to maximize the greenness) by dispatching the requests to proper DCs under the SLA and budget constraints. This paper presents optimization techniques for dynamic workload balancing for cloud-scale data center (DC) 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 DC workloads. Extensive evaluations support our solution's potential to minimize the environmental penalty for Internet service providers under the budget while fulfilling their SLAs.

1 INTRODUCTION

The awareness towards the reduction of the emission of green house gases (GHG) is increasing for the protection of global and local environments. 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 (Webb et al., 2008).

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 (Commission, 2013) which is already in its third phase. Importantly, in *cap-and-trade* the brown energy cap is reduced over time so that total emissions are progressively reduced with an ultimate goal of having zero emissions (Commission, 2013). This approach is being followed by industry (Google, 2011). To this end, a logical optimization goal is to maximize the usage of green energy, within bugetry constraints - the focus of this paper.

The most common instrument for trading in *cap-and-trade* schemes are Renewable Energy Credits (RECs): each REC represents one MWh of renewable energy contributed to the power grid. The facilities that produce this energy can be based on wind or solar farms. Importantly, RECs are not the same as energy. 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 (Google, 2011)). 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 (SolarBuzz,

Munawar W., Chen J. and Li M..
Minimizing Environmental Footprints of Data Centers under Budget and Service Requirement Constraints.
DOI: 10.5220/0004934202220232
In *Proceedings of the 3rd International Conference on Smart Grids and Green IT Systems* (SMARTGREENS-2014), pages 222-232
ISBN: 978-989-758-025-3

2013).

Data centers (DCs), being the biggest users of electricity in the IT sector (Paul Ontellini, 2011), have a significant environmental impact. One essential problem for managing DCs, according to the greenness tendency, is to minimize the environmental penalty by dispatching the requests to proper DCs under the service level agreements (SLAs) and budget constraints. There have been several results in the literature, e.g., (Zhang et al., 2011), (Shah et al., 2008), (Rao et al., 2010), (Le et al., 2010a), (Qureshi et al., 2009). Most of these researches ((Zhang et al., 2011), (Shah et al., 2008), (Rao et al., 2010), (Qureshi et al., 2009)) focus on the satisfying the average response time whereas actual SLAs often demand percentile guarantees. In (Le et al., 2010a), the percentile guarantees of SLAs are considered under the setting that the brown energy consumption is capped for each DC, whereas, a cap per enterprise is a more realistic model as discussed previously. In (Zhang et al., 2012), the authors consider the effect of DCs' demand on market prices of electricity. Detailed discussion about the related work follows in Section 8.

Our Contribution: This paper focuses on the minimization of the environmental footprint of DCs 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 DCs. This optimization problem is multifaceted by considering many important aspects in such a setting, explained in detail in Section 2. In our approach, we divide the problem into subproblems to be solved individually by each DC and by the central dispatching hub. We present a practical solution that encompasses all the energy-consuming components in a DC. That includes the energy consumption from the infrastructure for networking, computation, and cooling devices. Our solution is flexible enough to be applicable to DCs consisting of heterogeneous servers as well as able to accommodate different SLAs. We evaluate this with real-world workload traces from Wikipedia (Urdaneta et al., 2009) and varying electricity prices from different regions in USA obtained from NYISO (NYISO, 2013). 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.

2 BACKGROUND

This section presents the important aspects in achieving greenness in DCs.

Varying Price of Electricity. The price of both green and brown electricity vary temporally and geographically. Also, the variance in energy used for the requests, i.e., the active energy component, is a significant fraction of the total energy (Qureshi et al., 2009). Hence, an appropriate service placement can result in significant gains.

Multiple Services with Different SLAs. DCs are expected to offer more than one service to more than one client, under different SLAs and with different pricing. Majority of the previous work has focused on a single DC providing a single service. The impact of multiple SLAs and multiple services being offered by a group DCs has often not been considered.

Session-based Services. In the case of session-based services offered by DCs, not all requests can be arbitrarily routed to any DC. The requests belonging to one session must either be served by the same DC, or context transfer be quantified.

Communication Latency due to Geographical Distance. The geographical distance between the DCs and the front end causes additional delay in serving the routed requests (Qureshi et al., 2009). The effect of this delay on SLA should be considered when distributing requests.

Energy Cost of Sleep-wake Transitions. Putting a server in a DC 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. DCs 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 (Stansberry and Kudritzki, 2012)). 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 DC: (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. Many DCs nowadays include some local facilities to produce green energy, e.g., (Apple Inc., 2012), (Upson, 2007). The energy produced by the local facilities is audited and converted to carbon credits (NC-RETS, 2013) 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.

3 SYSTEM MODEL

In this section we formalize the system model and discuss how we handle the the challenges discussed previously (Section 2).

We consider a network of N DCs as shown in Fig 1. A central dispatcher receives all the requests and dispatches them to the N DCs 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 Λ_b (the request rate) in a control period b.

Energy Sources. We consider that each DC 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 DC i during control period b is $C_{b,i,j}$. We assume that $C_{b,i,j}$ is time varying. Importantly, fixed-cost energy contracts are just a special case of this more general setting. DCs with local green energy production facilities have to bear the initial investment and continuous management 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 DC i, the associated penalty is defined as $\phi_{i,j}$. In general, green energy sources have none, while brown energy source has a positive penalty.

The availability of renewable energy and carbon credits in the market depends on the 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 DCs is limited to maximum usage of L_j in the current budgeting period.

Service Level Agreements. DCs offer multiple services to multiple clients under different SLAs. This factor can be incorporated by dividing each DC into smaller cells to cover all the services that should be provided through the DCs. Each cell is considered as an individual DC. However, it is not that each DC 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 (Le et al., 2010b). (2) Not all clients are geographically suitably located to be served by some of the DCs because the communication latency is correlated to the geographical distance (Qureshi et al., 2009). Therefore the clients can be statically assigned to a subset of DCs. 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 (Zhao et al., 2000). *For the rest of this paper, we only present how to deal with one SLA for the simplicity of presentation.*

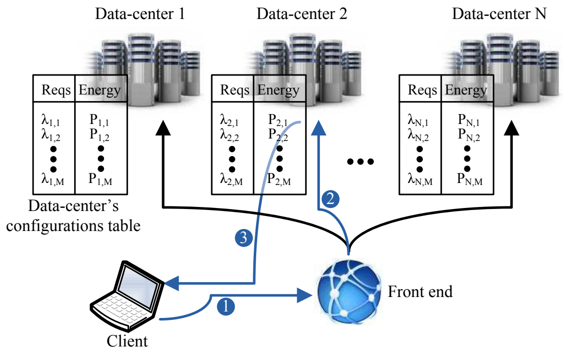


Figure 1: Arch. overview of a network of N data centers with a typical route for a request and its reply.

Session-based Services. Incoming requests from the clients are distributed by a central dispatcher. We assume that once a request has been routed, the reply comes directly from the corresponding DC. If it is a session-based service, all the further correspondence is directly with the DC where the first request in the session was assigned to. We assume that the front end is not part of the routing process after the initial decision hence does not cause any additional latency.

DC Configuration Table. Every DC 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 DC. This behavior can be captured in a table for energy requirement versus the maximum service (in terms of the request rate) in a DC under the specified SLA.

We consider DCs with discretized service levels, and each level has its required energy consumption in a control period. Every DC 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

DC 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 DC can be found in the literature, e.g. the methodologies in (Guerra et al., 2008) or (Chen et al., 2011). The energy consumed by infrastructure is also part of the total energy consumption $E_{i,k}$. *The DC 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 DC with selectable granularity.

Another important aspect is the energy cost for the off→on transitions of the servers in the DCs. We assume that the entries in a DC 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 (∼1 min (Le et al., 2010b)) 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 DC 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 DC 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.

4 PROBLEM DEFINITION AND FUTURE PREDICTION

4.1 Problem Statement

The objective is to *minimize the total environmental penalty* in the current budgeting period while satisfying the *service requirement* with the quality of service (QoS) as contracted in the SLA, without exceeding the *total budget S* with the time varying energy prices. Each DC 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 DC i such that the total environmental penalty is minimized under the service requirement constraint $\sum_{i=1}^{N} \lambda_{i,k_i} \geq \Lambda_b \; \forall b$ and the budget constraint.
Summarizing this,

i, j, k, b	=	Indices for DCs, energy sources, configurations and control periods
N	=	Total number of DCs
M	=	Max number of configurations per DC
Z	=	Maximum type of energy sources
B	=	Max control periods in budgeting period
L_j	=	Maximum energy availability from j^{th} source for all DCs combined (kWh)
S	=	total allowed cost budget for all DCs ($)
$E_{b,i,k}$	=	Energy required at DC i for k^{th} confguration during b^{th} control period (kWh)
$\phi_{i,j}$	=	penalty associated with j^{th} energy source in i^{th} DC (kg of CO_2)
$C_{b,i,j}$	=	cost of j^{th} energy source in i^{th} DC during the b^{th} control period ($ per kWh)
Λ_b	=	total service required during the b^{th} control period
$\lambda_{i,k}$	=	service provided at DC i's kth configuration
$x_{b,i,j}$	=	In i^{th} DC, 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. binary decision variables

With these symbols, the optimization problem can be formulated as follows:

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

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

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

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

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

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

These can be restated as:

1b: Usage of any energy source in a DC in any control period can not be more than total energy requirement for that data center in that control period.

1c: Sum of all the portions from all the energy sources should satisfy the energy requirements of the DC.

1d: Provided service should satisfy the required service for all control periods.

1e: Usage of any energy source cannot exceed its availability in the market.

1f: The sum of the costs occurring at the DCs should remain within the overall budget.

4.2 Infeasibility due to Unknown Future

A solution to the problem detailed in Equations (1a)-(1f) will result in the optimal reduction in environmental penalty. However, to solve this, we need Λ_b and $C_{b,i,j}$ for all future control periods. This is, however, not possible. Electricity prices change on hourly basis and the horizon for "certain" knowledge spans only an hour in future. Similarly, as service requests follow long term (monthly) and short term (hourly) trends (see Figure 3), good enough predictions are possible only for an hour in advance. Due to these factors we transform the problem to maximize the use green energy within a *single control period*. The problem can be modified as follows for a control period b, where $1 \leq b \leq B$: (with modified set of old symbols which belong only to a single control period)

$$\text{Minimize: } \sum_{i=1}^{N}\sum_{j=1}^{Z}\sum_{k=1}^{M} y_{i,k} \cdot E_{i,k} \cdot x_{i,j} \cdot \phi_{i,j}, \tag{2a}$$

$$\text{such that: } \quad 0 \leq x_{i,j} \leq 1, \quad \text{for all } i,j \tag{2b}$$

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

$$\sum_{i=1}^{N}\sum_{k=1}^{M} y_{i,k} \cdot \lambda_{i,k} \geq \Lambda, \tag{2d}$$

$$\sum_{i=1}^{N}\sum_{k=1}^{M} y_{i,k} \cdot x_{i,j} \cdot E_{i,k} \leq L_j - L_j^{b-1}, \quad \text{for all } j \tag{2e}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{Z}\sum_{k=1}^{M} y_{i,k} \cdot E_{i,k} \cdot x_{i,j} C_{i,j} \leq \psi(S - S^{b-1}). \tag{2f}$$

here,

- ψ = function for budget distribution. It must satisfy: $\psi(\Delta) \leq \Delta$. $\psi(\Delta)$ can be as simple as $\frac{\Delta}{B-b+1}$ or can be complex to include the predictions of traffic and pricing.
- L_j^{δ} = Used-up quota of energy availablilty for j^{th} type of energy upto δ control period, where $L_j^0 = 0$.
- S^{δ} = Budget consumed in the past for control periods upto δ with $S^0 = 0$.

Henceforth, we tackle the problem of greening the DCs as per Equations (2a)-(2f) i.e., according to the methodology shown in Figure 2. For every control period, we first calculate the budget on basis of traffic forecast. Predictions based on historical information or other prediction models, e.g., (Verma et al., 2010), (Box and Jenkins, 1994), 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 Λ constraint with the specified SLA. The requests are dispatched to different DCs as a result of the second step. *The main focus of our methodology in this paper is the second step, i.e. load balancing.* We assume that dispatching overhead is negligible.

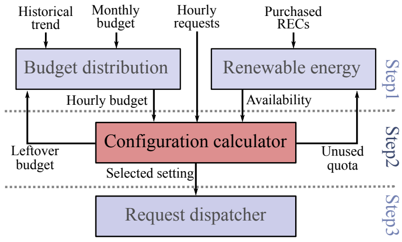


Figure 2: Outline for modified methodology.

Hardness. The problem formulated in Equations (2a)-(2f) is $\mathcal{NP}$-hard even for deriving a feasible solution. This can be proved by reducing from the decision version of the knapsack problem.

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 (Johnson and Garey, 1979).

The reduction works as follows: We construct N DCs such that each DC has only two configurations for the performance and energy consumption. That is, for DC 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. Hence, we conclude that deriving a feasible solution under budget and performance constraints for the studied problem is $\mathcal{NP}$-hard. □

5 OUR SOLUTION

The drawback of solving the optimization problem separately for each control period (Equations (2a)-(2f)), 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.

5.1 Dynamic Programming (DP)

5.1.1 Penalty Table for a DC

We first consider how to optimize for any DC i in a control period when the local budget S_i and the local service requirement Γ_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 $\lambda_{i,k^*} \geq \Gamma_i$.

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 DC i. It is now clear that the objective for this case is to minimize $E_{i,k^*} \sum_{j=1}^{Z} x_{i,j} \cdot \phi_{i,j}$ such that $\sum_{j=1}^{Z} x_{i,j} \cdot C_{i,j} \cdot E_{i,k^*} \leq S_i$ and $\sum_{j=1}^{Z} x_{i,j} = 1$. This can be solved by using the linear programming solver in general. Since, the green energy sources have zero 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 Γ_i, we can build the corresponding penalty table $p(i, \Gamma_i, S_i)$ to show the minimum penalty for DC i under the above configurations. If it is not feasible to support Γ_i under budget S_i, then, $p(i, \Gamma_i, S_i)$ will be set to ∞.

We remove the infeasible and dominated entries in the penalty p-table for DC 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 DC 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 DC i with $k \leq Q_i$, we denote

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

5.1.2 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 DC to provide the total required service.

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

$$P(1, \lambda, s) = p(1, \lambda, s). \tag{3}$$

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 $\lambda \geq 0$:

$$\begin{aligned} P(i, \lambda, s) \quad = \quad & \min_{k=1,2,\ldots,Q_i} \{P(i-1, \lambda - \ell_{i,k}, s - s_{i,k}) \\ & + \pi_{i,k}\}. \end{aligned} \tag{4}$$

Clearly, $P(N, \Lambda, S)$ 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 $P(N, \Lambda, S)$. The time complexity for calculating a single entry $P(i, \lambda, s)$ based on Equation (4) 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 Λ and from $s = 0$ to $s = S$ sequentially. This gives the overall time complexity $O(NS\Lambda Q_{\max})$, 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 λ from 0 to Λ 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 λ to the nearest integer multiple of a given number, let's say, I_λ. That is, $\ell'_{i,k}$ is $\left\lfloor \frac{\ell_{i,k}}{I_\lambda} \right\rfloor I_\lambda$. Then I_s and I_λ can serve as the discretization factors of budget S and Λ. This makes the time complexity to $O(N \frac{S}{I_s} \frac{\Lambda}{I_\lambda} Q_{\max})$.

5.2 Greedy Algorithm

We now present a heuristic algorithm based on a greedy strategy without building the penalty p-table constructed in Section 5.1.1. 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 DC i has been decided to use the k_i^{th} configuration. That is, it will provide λ_{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 DC i. If k_i is given for every DC i, the objective for this case is to

$$\text{minimize} \quad \sum_{i=1}^{N} E_{i,k_i} \sum_{j=1}^{Z} x_{i,j} \cdot \phi_{i,j} \tag{5a}$$

$$\text{such that} \quad \sum_{i=1}^{N} \sum_{j=1}^{Z} x_{i,j} \cdot E_{i,k_i} C_{i,j} \leq S, \tag{5b}$$

$$\sum_{j=1}^{Z} x_{i,j} = 1, \quad \text{for all } i \tag{5c}$$

$$\sum_{i=1}^{N} E_{i,k_i} \cdot x_{i,j} \leq L_j. \quad \text{for all } j \tag{5d}$$

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 DCs 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 (5a). If $\sum_{i=1}^{N} \lambda_{i,k_i}$ is no less than Λ, the algorithm terminates; otherwise it increases one DC i^* among the DCs 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 DC i to the configuration k_i+1, we can find the optimal setting in Equation (5a) for minimizing the penalty under this setting. Please note that the penalty is set to ∞ if there is no feasible solution for Equation (5a). By advancing the configuration of DC i, suppose that $\Delta_i^{service}$ is additional service, $\Delta_i^{penalty}$ is the additional penalty, and Δ_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 DCs, Service requirement: Λ, Budget: S, weights: w_b, w_e
Output: Configuration for all DCs: k_i

$k_i \longleftarrow 1$ for each DC i;
while *true* **do**
  if $\sum_{i=1}^{N} \lambda_{i,k_i} \geq \Lambda$ **then**
    if *Equation (5a) has a feasible solution* **then**
      return the solution k_i for each DC i with the purchase plan by solving Equation (5a) optimally;
    else
      return the solution k_i for each DC i but with "over budgeting" by buying all energy from the cheapest brown source;
  for *each DC i with* $k_i < M$ **do**
    $\Delta_i^{service} \longleftarrow \lambda_{i,k_i+1} - \lambda_{i,k_i}$;
    calculate Δ_i^{budget},$\Delta_i^{penalty}$ based on Equation (5a);
  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 DC 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 DCs and the number of iterations in the while loop is at most NM. For each combination, we have to solve Equation (5a). This can be sped up by starting based on the current solution. However, solving Equation (5a) by using linear programming solvers is already quite efficient. As we are not able to guarantee the budget satisfaction, over budgeting may be needed by borrowing from future invocations, as presented in pseudo-code.

5.3 Greedy + DP (G+D)

The greedy algorithm, 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 DCs 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: 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 $O(N^3 M \frac{S}{I_s} \frac{\Lambda}{I_\lambda} Q_{max})$, using the previously introduced symbols.

In the following sections we present our simulation setup and evaluation results.

6 SIMULATION SETUP

We adopt the settings from (Zhang et al., 2011) to evaluate the proposed solution by simulating the Google's setup for the location of DCs in the US. For these locations, we obtain the electricity pricing information from (NYISO, 2013). For our simulations, the following factors are important.

Non-varying Factors include the hardware capabilities of the DCs. These include server capabilities and cooling infrastructure. We consider four DCs, in which each data center is equipped with homogeneous servers, as detailed in Table 1. We use the method in (Wang et al., 2012) to build the DC configuration table, presented in Section 3, by considering 50 servers in each data center. The resulting table has at most 87 entries in each data center. Other methodologies like (Guerra et al., 2008) and (Chen et al., 2011) can also be adopted for calculating the DC configuration tables. Please note that the complexity of the presented solutions does not directly depend on the number of servers in DCs, but the number of entries in the DCs' configuration tables. Even when the servers in a DC increase, we can reduce the number of entries in the DC configuration tables by changing the management granularity.

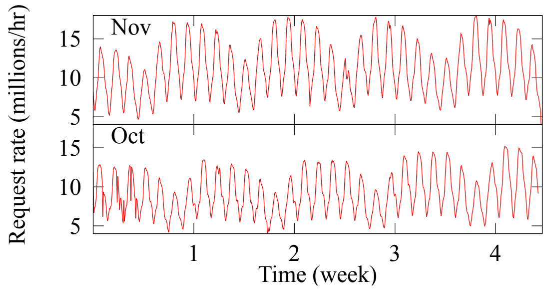


Figure 3: Wikipedia workload trace in Oct. and Nov. 2007.

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 environmental penalty.

Time-varying Factors include energy prices. The availability for both 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 is fulfilled. Green energy has a higher price than brown energy as explained in the introduction (Section 1). In our simulation we assume a surcharge of 1.5 cents and 18.0 cents per kWh for wind and solar energy (SolarBuzz, 2013) respectively in addition to the brown energy price. For price trace of electricity, we use the data from NYISO(NYISO, 2013). Specifically, we use Day-Ahead price data for Nov'07 for four regions previously mentioned.

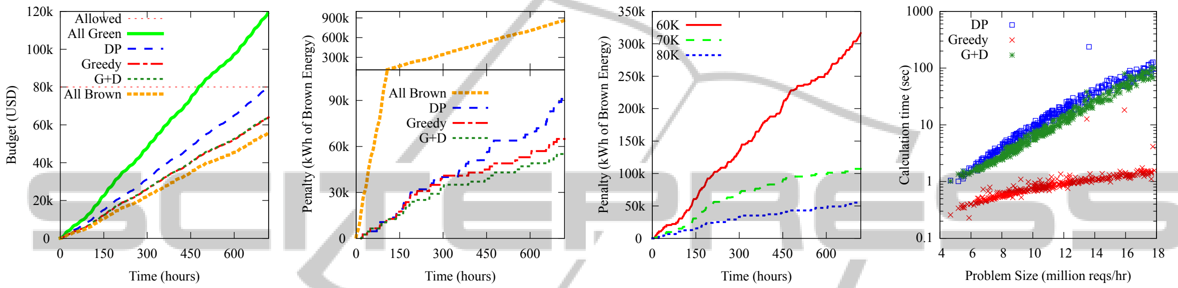
The other time varying factor is the total service requirement, Λ_b. It is a random variable but overall it follows a weekly recurring pattern (see Figure 3). We use the actual workload trace from Wikipedia (Urdaneta et al., 2009). We use Oct'07 for forecasting and the Nov'07 for the actual workload.

7 EVALUATION

In this section we present the results of our evaluations. We take a month as a budgeting period and an hour as a control period. For the greedy algorithm proposed in Section 5.2, 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 DP approach (Section 5.1) and simple greedy (Section 5.2).

Table 1: Data center settings used for simulation (adopted from (Li et al., 2012)): Speed ratio is the ratio of the frequency by adopting dynamic voltage frequency scaling (DVFS) to the maximum frequency in the system.

	DC # 1			DC # 2			DC # 3			DC # 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		
Power settings	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)
	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



(a) Budget usage. (b) Method comparison. (c) Budget vs. penalty (G+D). (d) Problem size vs. calculation time.

Figure 4: Evaluation Results.

7.1 Budget Allocation

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. Any other prediction scheme, e.g., (You and Chandra, 1999; Cortez et al., 2006) can be used for better predictions.

The budget usage comparison is presented in Figure 4(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.

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 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 Fig. 4(b).

The fundamental difference between G+D and the DP approach is the allocation of budget. Unlike DP, 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 DP approach. DP 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.

7.2 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 Λ_b and $C_{b,i,j}$. Irrespective of the prediction scheme, this results in deteriorated prediction quality hence affecting the solution quality.

One way to decrease computation time can be to compute the necessary tables offline. But, due to the time-varying factors like price and availability of energies, the offline tables will be huge and unpractical.

Online computation of solution at the beginning of every control period is the only viable option. Figure 4(d) presents the calculation time as a function of problem size for a single control period on a normal desktop machine (Intel i3, 6GB 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. With growing problem size, the computation times increases linearly for the greedy algorithm and exponentially for dynamic programming based solutions. This was expected as DP is pseudo-polynomial time algorithm. However, G+D, still fairs better than DP alone. The is because of the reduced search space due to the pruning by greedy algorithm.

8 RELATED WORK

DCs being major electricity consumers in the IT sector, have been focus of lot of research to make them environmental friendly. This can be divided into three main categories:

Energy Conservation: These studies aim to decrease the energy consumption of a DC, whereas decreased environmental footprint is a side product. Examples include (Chase et al., 2001), (Heo et al., 2007), (Wang et al., 2012). Mostly these aim to optimize a a single DC. For example Wang et al. (Wang et al., 2012) present a scheme to reduce power consumption while fulfilling the generalized SLAs within a single DC. The solution we present builds on top of these schemes as we aim for multiple DC optimization and single DC 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 DCs are considered. Examples in this category include (Qureshi et al., 2009), (Li et al., 2012), (Mathew et al., 2012), (Luo et al., 2013). Qureshi et al. (Qureshi et al., 2009) 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 (Li et al., 2012) and (Luo et al., 2013).

Utilizing the Green Energy: This is a relatively new direction with only few initial studies e.g (Zhang et al., 2011), (Shah et al., 2008), (Rao et al., 2010). Our approach falls in this category. (Zhang et al., 2011) present how to maximize the use of environmental friendly green energy to power the servers in DCs, 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 (Rao et al., 2010), (Le et al., 2009), (Shah et al., 2008). Moreover, (Rao et al., 2010) and (Shah et al., 2008) do not consider time-varying workloads, multiple services, or market interactions. Stewart and Shen (Stewart et al., 2009) 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. (Le et al., 2010a) is more thorough in their approach toward the problem. They focus on cost reduction by exploiting the distributed nature of DCs for dynamic request dispatching while maintaining SLAs. They are the first ones to consider carbon interactions. Our approach has two main differences from (Le et al., 2010a): 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.

9 CONCLUSION

The environmental footprint of DCs 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 DC. (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 DC.

The novelty of our approach lies in dividing the problem into two independent steps, that is, per DC optimization and a central optimization scheme. This forms the basis of general solution that can include factors like power consumption due to cooling infrastructure, power consumption of networking infrastructure, on-site renewable energy generation systems and multiple services with multiple SLAs.

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

REFERENCES

Apple Inc. (2012). Apple facilities environmental report. http://images.apple.com/environment/reports/docs/Apple_Facilities_Report_2012.pdf.

Box, J. and Jenkins, G. (1994). *Reinsel. Time Series Analysis, Forecasting and Control*.

Chase, J. et al. (2001). Managing energy and server resources in hosting centers. In *ACM SIGOPS Operating Systems Review*.

Chen, J.-J. et al. (2011). Power management schemes for heterogeneous clusters under quality of service requirements. In *SAC*.

Commission, E. (2013). The EU emissions trading system (EU ETS) - policies - climate action.

Cortez, P., Rio, M., Rocha, M., and Sousa, P. (2006). Internet traffic forecasting using neural networks. In *IJCNN'06*. IEEE.

Google (2011). Google's Green PPAs: What, How, and Why - R 02. *Google White Papers*.

Guerra, R. et al. (2008). Attaining soft real-time constraint and energy-efficiency in web servers. In *SAC*.

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

Johnson, D. and Garey, M. (1979). Computers and intractability: A guide to the theory of np-completeness. *Freeman&Co, San Francisco*.

Le, K., Bianchini, R., Martonosi, M., and Nguyen, T. (2009). Cost-and energy-aware load distribution across data centers. *HotPower*.

Le, K. et al. (2010a). Capping the brown energy consumption of internet services at low cost. In *Green Computing Conference*. IEEE.

Le, K. et al. (2010b). Managing the cost, energy consumption, and carbon footprint of internet services. In *ACM SIGMETRICS*.

Li, J. et al. (2012). Towards optimal electric demand management for internet data centers. *IEEE Transactions on Smart Grid*.

Luo, J. et al. (2013). Data center energy cost minimization: A spatio-temporal scheduling approach. In *INFOCOM*.

Mathew, V. et al. (2012). Energy-aware load balancing in content delivery networks. In *INFOCOM*, pages 954 –962.

NC-RETS (2013). North carolina renewable energy tracking system (NC-RETS). http://www.ncrets.org/.

NYISO (2013). New York Independent System Operator. http://www.nyiso.com/.

Paul Ontellini (2011). Intel CEO speaking at Dell World.

Qureshi, A. et al. (2009). Cutting the electric bill for internet-scale systems. In *ACM SIGCOMM*, pages 123–134.

Rao, L. et al. (2010). Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *INFOCOM*, pages 1–9. IEEE.

Shah, A. J. et al. (2008). 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.

SolarBuzz (2013). Cost-Competitiveness-Solarbuzz. http://solarbuzz.com/facts-and-figures/markets-growth/cost-competitiveness.

Stansberry, M. and Kudritzki, J. (2012). Data center industry survey. Technical report, Uptime Institute.

Stewart, C. et al. (2009). Some joules are more precious than others: Managing renewable energy in the datacenter. In *HotPower*.

Upson, S. (2007). The greening of google. *Spectrum, IEEE*, pages 24–28.

Urdaneta, G. et al. (2009). Wikipedia workload analysis for decentralized hosting. *Elsevier Comp. Networks*.

Verma, A. et al. (2010). Brownmap: Enforcing power budget in shared data centers. *ACM Middleware*, pages 42–63.

Wang, S. et al. (2012). Power-saving design for server farms with response time percentile guarantees. In *RTAS*, pages 273–284.

Webb, M. et al. (2008). Smart 2020: Enabling the low carbon economy in the information age. *The Climate Group. London*.

You, C. and Chandra, K. (1999). Time series models for internet data traffic. In *LCN'99*. IEEE.

Zhang, Y. et al. (2011). Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In *Middleware*.

Zhang, Y., Wang, Y., and Wang, X. (2012). Electricity bill capping for cloud-scale data centers that impact the power markets. In *ICPP*.

Zhao, W., Olshefski, D., and Schulzrinne, H. G. (2000). Internet quality of service: An overview.