Fuel Cell Generation in Geo-Distributed Cloud Services: A Quantitative Study

Zhi Zhou[1] Fangming Liu[1]* Bo Li[2] Baochun Li[3] Hai Jin[1] Ruolan Zou[1] Zhiyong Liu[4]

[1]Key Laboratory of Services Computing Technology and System, Ministry of Education,
School of Computer Science and Technology, Huazhong University of Science and Technology, China
[2]The Hong Kong University of Science and Technology [3]University of Toronto
[4]The Institute of Computing Technology, Chinese Academy of Sciences & The State Key Laboratory for Computer Architecture, ICT, CAS

***Abstract*—The demand for capping carbon emission has promoted the use of fuel cell energy in cloud computing, yet it is unclear what and how much benefit it may bring. This paper, for the first time, attempts to quantitatively examine the benefits brought by fuel cell generation, and to illustrate how such benefits can be realized with an intelligent coordination between grid power and fuel cell generation. Specifically, we propose UFC, a quantitative index called the *utility of the cloud using fuel cells*, which captures the level of the datacenters operator's overall satisfaction from energy cost, carbon emission, and workload performance. We formulate the UFC maximization problem to jointly optimize both fuel cell generation and geographical request routing. In order to avoid centralized solutions with high complexity and low scalability, we develop a distributed algorithm blending the advantages of Alternating Direction Method of Multipliers (ADMM) and the auxiliary variable method, whose performance is evaluated and verified through our extensive simulations based on real-world datacenter workload traces, electricity prices and generation data sets.**

I. Introduction

Internet-scale cloud services are deployed over geographically distributed datacenters, and are indispensable for a wide variety of applications, serving both enterprises and consumers. It is readily acknowledged, however, that their enormous and growing energy demand not only increases the total operational costs, but also has a profound environmental impact. It is estimated that datacenters will consume about 8% of the worldwide electricity by 2020, and produce 2.6% of the global carbon emission [1]. As one of the leading cloud service providers, Google emitted 1.68×10^6 tons of carbon in 2011 [2], 15.86% more than their emission in 2010, which is on par with the carbon emission of the United Nations [3].

Given the exacerbating pressure from both energy shortage and global warming, an increasing number of cloud service providers have started to incorporate renewable energy sources to reduce the carbon footprint of their infrastructures. By building wind farms and/or solar arrays to partially power their datacenters, prominent environmental benefits have been achieved. Recently, a new source of renewable energy, fuel cell generation with biogas, has attracted a tremendous amount of interest, as its technical advances have made it economical and feasible. For instance, Microsoft recently launched a project named "Data Plant" to run a datacenter powered by fuel cells with biogas from a nearby wastewater treatment plant [4]. Google recently announced that it would build a massive datacenter for Gmail and Google+ services, powered by energy coming from fuel cells that use biogas from livestock farms [5]. Providers such as Apple, eBay and Facebook are also investing in fuel cell generation to "clean up" their cloud services.

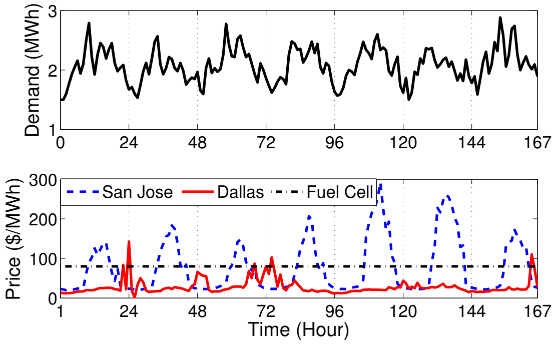

Fig. 1: The one-week Facebook power demand profile and the electricity prices at Dallas and San Jose, September 10 – 16, 2012.

There are several distinctive features that make fuel cell generation more appealing than wind energy and solar power in modern datacenters. *First,* fuel cells running on biogas can provide carbon neutral (*i.e.*, carbon-free) generation, with no environmental pressure to the service providers. *Second,* fuel cell generation may lead to cheaper power generation, reversing the recent trend of rising power generation costs. This implies that reducing the carbon footprint and energy cost are no longer mutually exclusive. *Finally and most importantly,* the salient advantage of fuel cell generation in the context of cloud computing is that, it allows the use of tunable output of fuel cells to track the time-varying workload, without compromising the workload performance. This is in sharp contrast with datacenters using intermittent wind/solar energy, where a performance scaling approach (*e.g.*, DFVS) is used to track the time-varying renewable power [6], or workload must be routed to datacenters with abundant renewable energy [7].

Though using fuel cells can potentially lead to substantial energy savings and significant environmental benefits, it

*The Corresponding Author is Fangming Liu (fmliu@hust.edu.cn). The research was supported in part by a grant from The National Natural Science Foundation of China (NSFC) under grant No.61370232, by a grant from National Basic Research Program (973 program) under grant No.2014CB347800; Zhiyong Liu's work is supported in part by China National Natural Science Foundation (NSFC) and Hong Kong RGC Joint Project No.61161160566, NSFC International Coordination Project No.61020106002, and NSFC Project for Innovation Groups No.61221062; Bo Li's work is supported in part by a grant from RGC under the contract 615613, by a grant from NSFC/RGC under the contract N_HKUST610/11, by a grant from NSFC under the contract U1301253, and by a grant from ChinaCache Int. Corp. under the contract CCNT12EG01.

TABLE I: Energy costs ($) of different strategies in Dallas and San Jose.

Strategy	Grid	Fuel Cell	Hybrid
Dallas	9644	27957	9387
San Jose	28470	27957	18250

remains unclear how such benefits can be materialized in geo-distributed cloud services. To show a high-level understanding of this problem, we evaluate the benefit of using fuel cells from the perspective of saving energy costs, by using the Facebook datacenter power demand profile [8], historical grid power prices and the current price for fuel cell generation, all of which are shown in Fig. 1. We consider three alternative strategies: (1) *Grid*, which only uses electricity from the power grid; (2) *Fuel Cell*, which only uses fuel cell generation; and (3) *Hybrid*, which dynamically switches between grid and fuel cell generation, and always chooses the cheaper alternative. The one-week energy cost of each strategy is summarized in Table I. The data in Dallas shows that completely relying on fuel cells may incur much higher energy costs, while data in both Dallas and San Jose demonstrate that a hybrid strategy that coordinates the use of grid power and fuel cell generation is most beneficial.

As the first dedicated effort, this paper attempts to unfold the benefits of fuel cell generation brought to the geo-distributed cloud, and to understand the role of fuel cell generation in the cloud. Specifically, we focus on the most significant potential benefit brought by fuel cell generation, and how it can be achieved through intelligent coordination between grid power and fuel cell generation. We first propose a quantitative index, referred to as the *utility of the cloud using fuel cells (UFC)*, which is significantly different from and complementary to the existing indexes such as PUE (Power Usage Efficiency) [9], CUE (Carbon Usage Efficiency) [10], and ERP (Energy-Response Time Product) [11]. We point out that UFC can be conceived as the degree of the datacenters operator's overall satisfaction from energy cost, carbon emission and workload performance in terms of propagation latency, which are typically the three most critical metrics to consider in datacenter operations. We formulate the maximization of UFC as a joint optimization problem on fuel cell generation and geographical request routing.

However, since the scale of our problem is prohibitively large, we propose to design a distributed control algorithm that is scalable, robust, and amenable to practical implementations, based on *Alternating Direction Method of Multipliers (ADMM)*. ADMM [12] works well for linearly constrained convex problems whose objective function is separable into m $(m \geq 1)$ individual convex functions with non-overlapping variables. It alternatively optimizes part of the objective with one block of variables to reach the optimum with fast convergence. Our problem is a 4-block ADMM problem, however, the convergence of conventional ADMM for $m \geq 3$ is only achievable with the assumption that the objective function is strongly convex [13]. Since our work attempts to examine the effectiveness of current carbon tax function that may not be strongly convex in promoting renewable energy, we apply a variant of ADMM, called ADM-G [14], without relying on this assumption. Furthermore, instead of directly applying the original ADM-G that leads to a centralized solution, we develop an efficient distributed algorithm by blending the advantage of the auxiliary variable method and ADM-G.

II. The Utility Maximization Model

In this section, we propose UFC to quantify the potential benefit brought by fuel cell generation, and formulate the UFC maximization problem to jointly optimize both fuel cell generation and geographical request routing.

A. System Overview

We consider a provider running cloud services on N geographically distributed datacenters, denoted by $\mathcal{D} = \{1, 2, ..., N\}$, and each datacenter $j \in \mathcal{D}$ consists of S_j homogeneous servers. Although we assume that all the servers at one datacenter are homogeneous, note that our model is quite general and can be easily extended to capture the heterogeneous case with a few additional notations. The cloud deploys M front-end proxy servers, denoted by $\mathcal{S} = \{1, 2, ..., M\}$ in various regions to direct user requests to the appropriate datacenters.

In progressive datacenters operated by Google, Apple, Microsoft, *etc.*, fuel cell generation (*e.g.*, Bloom Energy Server [15]) has been massively and increasingly deployed to cap their carbon footprint. To capture this tendency, we assume that each datacenter $j \in \mathcal{D}$ deploys fuel cells using biogas to generate carbon-free power [15], with maximal electricity output capacity $\mu_j^{\max}$.

Inspired by the latest modeling work on datacenter [16], [17], we adopt a discrete time-slotted model where the time slot matches the time scale at which the fuel cell generation and geographical request routing decisions are updated. At each time slot $t = (0, 1, 2, \cdots)$, an amount of user requests $A_i(t)$ from a common geographical region i arrive and aggregate at the front-end proxy server i. In practice, the near-term request arrival at each front-end proxy server can be predicted quite accurately, by employing techniques such as statistical machine learning and time series analysis [18]. Here we mainly consider interactive workloads (*e.g.*, web search) that can not be *deferred* and must be served *immediately*. Thus, the decision making is uncorrelated from slot to slot and we can focus on a single time slot for the analysis henceforth.

Though fuel cell generation has recently received prime interest in geo-distributed datacenters, there is no consensus on what and how much benefit it can bring. In order to remove these doubts, we first propose a quantitative index, referred to as the *utility of the cloud using fuel cells (UFC)*, to capture the benefit of using fuel cells in geo-distributed datacenters.

B. Utility of the Cloud Using Fuel Cells (UFC)

For modern geo-distributed datacenters that host diverse cloud services, cutting the rising electricity bill and carbon footprint without workload performance degradation is essential. On the other hand, fuel cell generation promises advantages of reducing both the energy cost and carbon footprint, as well as enabling tunable output to guarantee workload performance for geo-distributed cloud services.

Therefore, we propose the notion of UFC to quantitatively measure the degree of the cloud operator's overall satisfaction incorporating three components: workload performance, energy cost and carbon emission after using fuel cell generation. UFC captures the operator's target on workload performance

TABLE II: Key Parameters in the Utility Maximization Model.

Notations	Definitions
M	The number of front-end proxy servers $\mathcal{S}$
N	The number of geo-distributed datacenters $\mathcal{D}$
A_i	The amount of arrived requests at the front-end server i
S_j	The number of homogeneous servers in datacenter j
λ_{ij}	The amount of requests routed from front-end proxy server i to datacenter j
L_{ij}	The propagation latency between front-end proxy server i and datacenter j
μ_j	The fuel cell generation in datacenter j
ν_j	The power draw from the electricity grid in datacenter j
p_j	The electricity price at datacenter j
p_0	The price of fuel cell generation
C_j	The electricity carbon emission rate at datacenter j

improvement, energy cost saving and carbon emission reduction. Note that the interpretation of UFC is quite generic in realistic cloud services. In particular, UFC can be interpreted as the operator's total monetary payoff if we transform the carbon emission to monetary costs (*e.g.*, in the form of carbon taxes), and associate the workload performance with the economic gain.

Compared with the existing adopted indexes such as PUE (Power Usage Efficiency) [9], CUE (Carbon Usage Efficiency) [10], and ERP (Energy-Response Time Product) [11], UFC is significantly different from and complementary to them in at least two important aspects. First, indexes such as PUE and CUE focus on the efficiency of a single datacenter, while our proposed UFC is applicable to geo-distributed datacenters. Second, the existing indexes are proposed for specific functions, and none of them is capable of revealing the fuel cell generation's holistic efficiency in workload performance improvement, energy cost saving and carbon emission reduction.

To characterize each component of UFC, we identify the following two interrelated control decisions that can influence them: (i) *Fuel cell generation*, to determine the fuel cell output μ_j in each datacenter j, with the output capacity constraint:

$$0 \leq \mu_j \leq \mu_j^{\max}, \quad \forall j \in \mathcal{D},$$

where $\mu_j^{\max}$ is the aforementioned maximal output capacity of fuel cells in datacenter j; and (ii) *Geographical request routing*, to determine λ_{ij}, the amount of requests routed from the front-end proxy server i to datacenter j, with the following load balance and capacity constraints:

$$\sum_{j \in \mathcal{D}} \lambda_{ij} = A_i, \quad \forall i \in \mathcal{S},$$

$$\sum_{i \in \mathcal{S}} \lambda_{ij} \leq S_j, \quad \forall j \in \mathcal{D},$$

where A_i is the request arrival in terms of the number of servers required at the front-end proxy server i, and S_j denotes the number of servers in datacenter j defined in Sec. II-A.

With the control decisions above, we model each component of the UFC as follows, along with key notations in Table II:

1) ***Energy cost****:* Here, we focus on the power consumption of the servers in our energy cost model. Specifically, it has been widely shown by recent empirical studies [9] that, the aggregated power consumption of homogeneous servers can be measured as a linear function of the total workload, $SP_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}})\lambda$. S and λ denote the number of running servers and the amount of workload, respectively.

Based on the server power model above, for datacenter $j \in \mathcal{D}$ that hosts S_j homogeneous active servers and manages a workload of $\sum_{i \in \mathcal{S}} \lambda_{ij}$ and possesses a power usage efficiency of PUE$_j$, its server power consumption can be calculated as: $S_j P_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}}) \sum_{i \in \mathcal{S}} \lambda_{ij}$. Further, the total power demand at datacenter j can be given as:

$$D_j\Big(\sum_{i \in \mathcal{S}} \lambda_{ij}\Big) = \Big(S_j P_{\text{idle}} + (P_{\text{peak}} - P_{\text{idle}}) \sum_{i \in \mathcal{S}} \lambda_{ij}\Big) \cdot \text{PUE}_j.$$

The power usage efficiency metric PUE represents the ratio of the total amount of power used by the entire datacenter facility to the power delivered to the computing equipment. Inefficient datacenters can have a PUE $\in [2.0, 3.0]$, while leading industry datacenters are known to approach a PUE of around 1.1 [9].

For better readability and to highlight the relationship between the power demand and the amount of workload, we rewrite the equation above as: $D_j\big(\sum_{i \in \mathcal{S}} \lambda_{ij}\big) = \alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij}$, where $\alpha_j = S_j P_{\text{idle}} \cdot \text{PUE}_j$ and $\beta_j = (P_{\text{peak}} - P_{\text{idle}}) \cdot \text{PUE}_j$. Then, given the fuel cell generation μ_j which is no larger than the energy demand $\alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij}$, the power draw from the electricity grid can be given as: $\alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij} - \mu_j$. With the grid electricity price p_j and fuel cell generation price p_0, the total energy cost at datacenter j can be captured by: $p_j(\alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij} - \mu_j) + p_0 \mu_j$.

Note that the grid electricity p_j typically fluctuates over time [19], while the price of fuel cell generation p_0 is fixed. This observation implies that we can arbitrage the price difference by switching between the power grid and fuel cell generation. The total energy cost of the cloud is given by:

$$\sum_{j \in \mathcal{D}} \Big\{ p_j \big(\alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij} - \mu_j\big) + p_0 \mu_j \Big\}.$$

2) ***Carbon emission****:* The enormous carbon emission of geo-distributed clouds has been a vital concern for the providers. However, such a concern can be defused by deploying fuel cells in datacenters, since the fuel cell generation with biogas is carbon-free. Thus, given the j-th datacenter's power draw $\alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij} - \mu_j$ from the electricity grid, the carbon emission of datacenter j can be calculated as: $E_j = (\alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij} - \mu_j) C_j$. Here C_j is the electricity carbon emission rate, *i.e.*, the amount of carbon emission when generating 1KWh of electricity in location j. It can be estimated by summing the weighted contribution from each electricity fuel type as follows:

$$C_j = \frac{\sum_k e_{kj} \times c_k}{\sum_k e_{kj}}, \tag{1}$$

where e_{kj} represents the amount of electricity generated from fuel type k in location j, and c_k is the carbon emission rate of fuel type k given in Table III.

The real-time or historical electricity generation data that contains the fuel mix can be downloaded from the Regional Transmission Organization (RTO) website. Note that, as with the electricity price, the electricity carbon emission rate also exhibits both spatial and temporal diversity. Furthermore, previous research [20] has observed that the carbon emission rate at each region shows a strong diurnal pattern, making it easy to be accurately predicted.

TABLE III: Carbon dioxide emission per kilowatt-hour for the most common fuel types [1].

Fuel Type	**Nuclear**	**Coal**	**Gas**	**Oil**	**Hydro**	**Wind**
CO_2 g/kWh	15	968	440	890	13.5	22.5

In reality, except for the involvement of renewable energy, many cloud service providers are also willing or enforced to further offset the impact of their emission in the following monetary ways: (1) **Carbon tax**, in countries like Australia, Denmark and Norway, the heavy polluters will be taxed for the carbon emission with a fixed rate. For example, the datacenters among the top 500 polluters in Australia should pay \$23AUD for each tonne of carbon emitted. (2) **Cap and trade**, in the European Union, the cap is allocated to firms in the form of emission permits, which represent the right to emit a specific volume of carbon. Firms that need to increase their carbon emissions must buy permits from those who need fewer permits. (3) **Carbon offset**, the polluters can offset their emissions through investments on sustainable projects such as tree-planting to absorb carbon emission, or renewable energy certificates (RECs) to promote the development of green energy. These projects would incur an additional emission cost at datacenter j, denoted as $V_j(E_j)$. In practice, V_j has different forms at different locations or under different policies. In this paper, we only need to assume that V_j is a non-decreasing and convex function.

3) ***Workload performance:*** Arguably, latency is the most critical performance metric for interactive cloud applications, such as web search and social networking services. In this paper, we focus on the wide-area network propagation latency from a front-end proxy server to a processing datacenter, as it largely accounts for the user-perceived latency and overweighs other factors such as queuing or processing delays in datacenters [16].

The propagation latency L_{ij} between the front-end server i and datacenter j can be obtained through active measurements or other means in practice. Further, empirical studies have demonstrated that, the propagation latency L_{ij} can be approximated with geographical d_{ij} distance between the front-end server i and datacenter j as: $L_{ij} = d_{ij} \times 0.02$ms/km [9]. That is, 1 km in geographical distance incurs a propagation latency of approximately 0.02 ms.

As we have mentioned in Sec. I, for datacenters that incorporate intermittent wind/solar energy, the geographical mismatch between the workload demand and renewable supply would incur an extended workload propagation latency. However, since the fuel cells enable the capability of *load following* [21], *i.e.,* the use of tunable output to track the time-varying workload, the workload can be routed to datacenters nearby, and thus the propagation latency is reduced.

The utility of the user group that is served by the front-end proxy server i depends on the experienced average propagation latency $\sum_{j\in\mathcal{D}} \lambda_{ij}L_{ij}/A_i$ through a generic utility function U. U can take various forms depending on the cloud services and providers. Here we only assume that U is a decreasing and concave function. A commonly adopted utility function is a quadratic function that reflects the user's increased tendency to leave the service with an increased latency [1], [16], which is shown as follows:

$$U(\lambda_i) = -A_i\left(\frac{\sum_{j=1}^{N}\lambda_{ij}L_{ij}}{A_i}\right)^2, \tag{2}$$

where $\lambda_i = (\lambda_{i1}, \lambda_{i2}, \cdots, \lambda_{iN})^T$.

C. Maximization of UFC

Having quantified each component of the integrated utility UFC, we are now in a position to formulate the UFC maximization as an optimization problem that maximizes the total utility of serving user requests, minus the carbon emission and energy costs incurred.

$$\begin{aligned}
\max \quad & w\sum_{i\in\mathcal{S}} U(\lambda_i) - \sum_{j\in\mathcal{D}} V_j\Big(C_j\big(\alpha_j + \beta_j\sum_{i\in\mathcal{S}}\lambda_{ij} - \mu_j\big)\Big) \\
& -\sum_{j\in\mathcal{D}}\Big\{p_j\big(\alpha_j + \beta_j\sum_{i\in\mathcal{S}}\lambda_{ij} - \mu_j\big) + p_0\mu_j\Big\} && (3)\\
\text{s.t.} \quad & \sum_{j\in\mathcal{D}}\lambda_{ij} = A_i, \forall i \in \mathcal{S}, && (4)\\
& \sum_{i\in\mathcal{S}}\lambda_{ij} \le S_j, \forall j \in \mathcal{D}, && (5)\\
& \alpha_j + \beta_j\sum_{i\in\mathcal{S}}\lambda_{ij} - \mu_j \ge 0, \forall j \in \mathcal{D}, && (6)\\
\text{var} \quad & \lambda_{ij} \ge 0, 0 \le \mu_j \le \mu_j^{\max}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}.
\end{aligned}$$

Here, w is the weight of the workload latency compared to the total cost of carbon emission and energy, and it rests on the provider's pursuit. (4) and (5) are the aforementioned load balance and datacenter capacity constraints, respectively. (6) is the power balance constraint, which ensures that the power draw from the electricity grid is non-negative.

Remark: In the above formulation, we assume that all the servers at each datacenter are powered on and thus the parameter $S_j(\forall j \in \mathcal{D})$ is fixed. The rationale is that for commercial cloud services such as Amazon, reliability is more of a concern than shutting down the idle servers [22]. Nevertheless, our model is quite general and can be easily extended to incorporate the choice of shutting down the idle servers. Specifically, if we use $S_j^{\max}$ to denote the number of total available servers in datacenter j, then, we further need to determine the number of active servers S_j which satisfies the additional constraint $S_j \le S_j^{\max}, \forall j \in \mathcal{D}$ in each datacenter j.

Unfortunately, this maximization problem (3) has a very large scale, as the cloud may deploy tens of geo-distributed datacenters and hundreds of thousands of front-end proxy servers around the world. The scale of this problem motivates us to design a scalable and robust distributed algorithm that is amenable to practical implementations.

III. Distributed Algorithm

To address the challenge of solving the optimization problem (3) at a large scale, we resort to the alternating direction method of multipliers (ADMM) optimization technique, whose performance in solving large scale convex problems has been extensively studied [12]. However, rather than directly taking the ADMM technique for a centralized solution, we incorporate the advantages of both auxiliary variables and ADMM to design a new distributed control algorithm that maximizes the UFC.

A. A Primer on ADMM

ADMM solves the problem of minimizing a separable convex function subject to linear equality constraints:

$$\min \quad \sum_{i=1}^{m} f_i(x_i), \qquad \text{s.t.} \quad \sum_{i=1}^{m} K_i x_i = b, \tag{7}$$

with variables $x_i \in \mathbb{R}^{n_i}$ $(i = 1, ..., m)$, where $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ $(i = 1, ..., m)$ are closed proper convex functions, $K_i \in \mathbb{R}^{l \times n_i}$ are relation matrices, and $b \in \mathbb{R}^l$ is a relation vector. Note that the model (7) can easily accommodate general linear inequality constraint $\sum_{i=1}^{m} K_i x_i \leq b$ by adding one extra block. In particular, we can introduce a slack variable $x_{m+1} \geq 0$ and rewrite the inequality constraint as $\sum_{i=1}^{m} K_i x_i + x_{m+1} = b$.

A popular approach to the separable convex optimization (7) is to form the augmented Lagrangian by introducing an extra $L-2$ norm term $\| \sum_{i=1}^{m} K_i x_i - b \|^2$ to the objective:

$$\begin{aligned} \mathcal{L}_\rho(x_1, ..., x_m; y) = \sum_{i=1}^{m} f_i(x_i) + y^T (\sum_{i=1}^{m} K_i x_i - b) \\ + \frac{\rho}{2} \| \sum_{i=1}^{m} K_i x_i - b \|_2^2, \end{aligned} \tag{8}$$

where $\rho \geq 0$ is the penalty parameter. Clearly the augmented Lagrangian can be viewed as the unaugmented Lagrangian adding the penalty term, thus the minimization of $\mathcal{L}_\rho(x_1, ..., x_m; y)$ is equivalent to the original problem (7).

A generalized ADMM algorithm updates each x_i $(i = 1, 2, \cdots, m)$ in an alternating or sequential method as follows:

$$\begin{aligned} x_i^{k+1} &= \arg\min_{x_i} \mathcal{L}_\rho(x_1^{k+1}, ..., x_{i-1}^{k+1}, x_i, x_{i+1}^k, ..., x_m^k; y^k), \\ y^{k+1} &= y^k + \rho(\sum_{i=1}^{m} K_i x_i^{k+1} - b). \end{aligned} \tag{9}$$

When there is only one block, *i.e.*, $m = 1$, then ADMM reduces to the standard augmented Lagrangian method of multipliers, for which the global convergence is well understood. Further, when there are two blocks ($m = 2$), it has also been demonstrated that every limit point of the iterate is an optimal solution of the problem under mild assumptions. However, when there are more than two blocks ($m \geq 3$), the convergence of ADMM has long remained an open question.

Some recent progresses have been made to establish the global convergence of ADMM with $m \geq 3$. He *et al.* [14] proposed an algorithm called ADM-G, and established its provable convergence (but with unknown convergence speed) by correcting the output of ADMM with a Gaussian back substitution procedure. More recently, Xu *et al.* [13] proposed a distributed 4-block ADMM that is proven to converge with linear convergence speed, based on the assumption that each objective function f_i is strongly convex.

As our objective function may not be strongly convex, we choose to adopt ADM-G to devise our solution. In essence, ADM-G can be regarded as a *prediction-correction* scheme. Specifically, ADM-G predicts the new iterate in the forward order ($x_1^{k+1} \rightarrow \cdots \rightarrow x_m^{k+1} \rightarrow y^{k+1}$), and then corrects the predicted output in the backward order ($y^{k+1} \rightarrow x_m^{k+1} \rightarrow \cdots \rightarrow x_1^{k+1}$) via a Gaussian back substitution procedure. The detailed description of ADM-G is given as follows:

Step 1. ADMM step (prediction step). Obtain $(\widetilde{x}_1^k, \widetilde{x}_2^k, \cdots, \widetilde{x}_m^k, \widetilde{y}^k)$ in the forward (alternating) order:

$$\begin{aligned} \widetilde{x}_i^k &= \arg\min_{x_i} \mathcal{L}_\rho(\widetilde{x}_1^k, \cdots, \widetilde{x}_{i-1}^k, x_i, x_{i+1}^k, \cdots, x_m^k); \\ \widetilde{y}^k &= y^k + \rho(\sum_{i=1}^{m} K_i \widetilde{x}_i^k - b). \end{aligned}$$

Step 2. Gaussian back substitution step (correction step). Generate the new iterate $(x_1^{k+1}, x_2^{k+1}, \cdots x_m^{k+1}, y^{k+1})$ by correcting $(\widetilde{x}_1^k, \widetilde{x}_2^k, \cdots, \widetilde{x}_m^k, \widetilde{y}^k)$ in the backward order:

$$\begin{aligned} G(z^{k+1} - z^k) &= \varepsilon(\widetilde{z}^k - z^k); \\ x_1^{k+1} &= \widetilde{x}_1^k. \end{aligned} \tag{10}$$

Here $z = (x_2, x_3, \cdots, x_m, y)$, $\varepsilon \in (0.5, 1]$ is a constant, and G is the Gaussian back substitution matrix given as follows:

$$\begin{pmatrix} I_{n_2} & (K_2^T K_2)^{-1} K_2^T K_3 & \cdots & (K_2^T K_2)^{-1} K_2^T K_m & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \ddots & I_{n_{m-1}} & (K_{m-1}^T K_{m-1})^{-1} K_{m-1}^T K_m & 0 \\ 0 & \cdots & 0 & I_{n_m} & 0 \\ 0 & \cdots & 0 & 0 & I_l \end{pmatrix}$$

Insight: The matrix G defined above is an upper-triangular block matrix. The Gaussian back substitution step is thus easy to execute. In fact, as we have mentioned, after the predictor is generated by the ADMM scheme in the forward (alternating) order, the proposed Gaussian back substitution step corrects the predictor in the backward order.

The optimality and convergence of ADM-G can be guaranteed under very mild technical assumptions.

***Theorem** 1:* Assume that the matrices $K_i^T K_i$ $(i = 2, 3, \cdots, m)$ are nonsingular and the solution set of optimization (7) is nonempty, the sequence generated by the proposed ADM-G converges to an optimal solution of the problem (7).

B. Transforming the UFC Maximization to the ADMM Form

The UFC maximization problem (3) cannot be readily solved using ADMM, due to the following two facts. *First*, the fuel cell generation provisioning decision μ_j and the geographical request routing decision λ_{ij} are connected by an inequality constraint, rather than an equality constraint required by ADMM. *Second*, the carbon emission cost function in the objective function V_j couples both the decision μ_j and the decision λ_{ij}, and it is inseparable unless V_j is linear.

In response to these challenges, we introduce an additional block of variables $\nu_j = \alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij} - \mu_j$, and thus to separate the integrated utility function and transform the inequality constraint (6) to the equality constraint

$$\alpha_j + \beta_j \sum_{i \in \mathcal{S}} \lambda_{ij} - \mu_j - \nu_j = 0 \tag{11}$$

simultaneously. Note that $\nu_j \geq 0$, and it equals to the power draw from the electricity grid in datacenter j. The UFC maximization problem (3) is now transformed to the following problem in ADMM form:

$$\min \quad \sum_{j\in\mathcal{D}} \Big(V_j(C_j\nu_j) + p_j\nu_j + p_0\mu_j \Big) - w\sum_{i\in\mathcal{S}} U(\lambda_i) \quad (12)$$

$$\begin{aligned} \text{s.t.} \quad & (4), (5) \\ & \alpha_j + \beta_j \sum_{i\in\mathcal{S}} \lambda_{ij} - \mu_j - \nu_j = 0, \forall j \in \mathcal{D}, \\ \text{var} \quad & \lambda_{ij}, \nu_j \geq 0, 0 \leq \mu_j \leq \mu_j^{\max}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}. \end{aligned}$$

Is ADMM directly applicable? However, directly applying ADMM to problem (12) will lead to a centralized algorithm with high complexity, since the workload utility $\sum_{i\in\mathcal{S}} U(\lambda_i)$ couples λ_{ij}'s across j, while the penalty term $\sum_{j\in\mathcal{D}} \big(\alpha_j + \beta_j \sum_{i\in\mathcal{S}} \lambda_{ij} - \mu_j - \nu_j\big)^2$ and the capacity constraint $\sum_{i\in\mathcal{S}} \lambda_{ij} \leq S_j$ couples λ_{ij}'s across i. Thus, the workload utility ought to be separated from the penalty term and the capacity constraints if we need a distributed algorithm.

To this end, we continue to introduce a set of auxiliary variables $a_{ij} = \lambda_{ij}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}$, and reformulate the problem (12) as follows:

$$\min \quad \sum_{j\in\mathcal{D}} \Big(V_j(C_j\nu_j) + p_j\nu_j + p_0\mu_j \Big) - w\sum_{i\in\mathcal{S}} U(\lambda_i) \quad (13)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{j\in\mathcal{D}} \lambda_{ij} = A_i, \forall i \in \mathcal{S}, \\ & \sum_{i\in\mathcal{S}} a_{ij} \leq S_j, \forall j \in \mathcal{D}, && (14) \\ & \alpha_j + \beta_j \sum_{i\in\mathcal{S}} a_{ij} - \mu_j - \nu_j = 0, \forall j \in \mathcal{D}, && (15) \\ & a_{ij} = \lambda_{ij}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}, && (16) \\ \text{var} \quad & \lambda_{ij}, a_{ij}, \nu_j \geq 0, 0 \leq \mu_j \leq \mu_j^{\max}, \forall i \in \mathcal{S}, \forall j \in \mathcal{D}. \end{aligned}$$

Insight: The problem (13) is equivalent to the problem (12), where λ_{ij} controls the workload utility with only the load balance constraint (4), while a_{ij} ensures that the datacenter capacity constraint (14) and the power balance constraint (15) are enforced when the coupling happens across the front-end server i. This is the key idea that enables both the λ and a minimization to be decomposable, as we will demonstrate in the following.

The augmented Lagrangian $\mathcal{L}_\rho$ of problem (13) can be readily obtained from (8). By omitting the irrelevant terms, at each iteration $k+1$, the λ-minimization step involves solving the following problem according to (9):

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{S}} \Bigg(\sum_{j\in\mathcal{D}} \Big(\varphi_{ij}^k \lambda_{ij} + \frac{\rho}{2}(\lambda_{ij}^2 - 2a_{ij}^k\lambda_{ij}) \Big) - wU(\lambda_i) \Bigg) \\ \text{s.t.} \quad & \sum_{j\in\mathcal{D}} \lambda_{ij} = A_i, \lambda_{ij} \geq 0, \forall i \in \mathcal{S}, \end{aligned}$$

where φ_{ij} is the dual variable for the equality constraint (16).

Insight: The problem described above is clearly *decomposable* over i into M per-front-end server sub-problems, since the objective function and constraint are separable over i. Similarly, we also find that the minimization step for μ, ν and a are decomposable over datacenters. Therefore, the ADMM steps in our problem can be efficiently computed in a *fully distributed* manner.

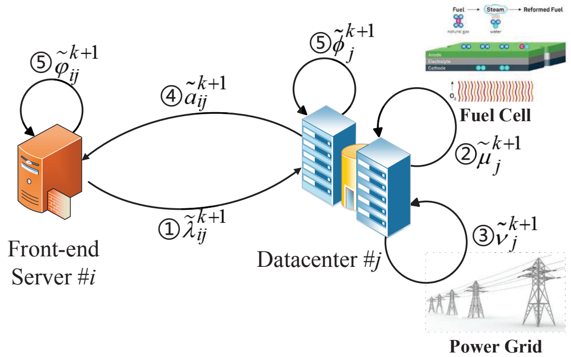

Fig. 2: An illustration of the information interaction between the front-end proxy servers and the back-end datacenters of each procedure in the ADMM step of our proposed distributed 4-block ADM-G algorithm.

C. Distributed UFC Maximization

In our 4-block ADMM problem (13), the workload utility function $U(\cdot)$ and carbon emission cost function $V_j(\cdot)$ can have various forms and *may not be strongly convex*. For instance, the current "cap and trade" policy in the real world is based on linear pricing, while the stepped tax system is also widely adopted in economic fields. Therefore, a direct application of the generalized ADMM (9) technique may not guarantee convergence.

To address this challenge, we transform the constraints $\lambda_{ij} = a_{ij}$ and $\alpha_j + \beta_j \sum_{i\in\mathcal{S}} a_{ij} - \mu_j - \nu_j = 0$ to the following standard form:

$$K_1\lambda + K_2\mu + K_3\nu + K_4a = b,$$

where $\lambda = (\lambda_{11}, \cdots, \lambda_{1N}, \cdots, \lambda_{M1}, \cdots, \lambda_{MN})^T$, $\mu = (\mu_1, \cdots, \mu_N)^T$, $\nu = (\nu_1, \cdots, \nu_N)^T$, $a = (a_{11}, \cdots, a_{1N}, \cdots, a_{M1}, \cdots, a_{MN})^T$ are the 4 blocks of variables, b is the relation vector, and K_i $(i = 1, 2, 3, 4)$ are the relation matrices.

Interestingly, we find that $K_i^T K_i$ $(i = 2, 3, 4)$ are nonsingular, which is just the case that ADM-G assumes. Before giving the detailed scheme of our distributed 4-block ADM-G, we first provide an intuitive and concrete understanding of the mechanism of our algorithm. Specifically, we illustrate the information interaction between the front-end proxy servers and the back-end processing datacenters of the 5 procedures of the ADMM step in Fig. 2. For each iteration $k \geq 0$:

1) Based on the tuple $(a_{ij}^k, \varphi_{ij}^k)$, each front-end server i computes the predicted optimal request routing $\widetilde{\lambda}_{ij}^k$ and sends it to the corresponding datacenter j.
2) Each datacenter j computes the predicted optimal fuel cell generation $\widetilde{\mu}_{ij}^k$ based on the triple $(a_{ij}^k, \nu_{ij}^k, \phi_j^k)$.
3) Each datacenter j computes the predicted optimal power draw $\widetilde{\nu}_{ij}^k$ based on the triple $(\widetilde{\mu}_{ij}^k, a_{ij}^k, \phi_j^k)$.
4) Based on the quintuple $(\widetilde{\lambda}_{ij}^k, \widetilde{\mu}_{ij}^k, \widetilde{\nu}_{ij}^k, \phi_j^k, \varphi_{ij}^k)$, each datacenter j computes the predicted optimal request routing $\widetilde{a}_{ij}^k$ and sends it to the corresponding front-end server i.

5) Each front-end server i and datacenter j compute the predicted optimal dual variables $\widetilde{\varphi}_{ij}^k$ and $\widetilde{\phi}_j^k$, respectively.

The detailed iterative scheme of our proposed distributed 4-block ADM-G algorithm is as follows:

Distributed 4-block ADM-G. Initialize $\lambda, \mu, \nu, a, \phi, \varphi$ to 0. For $k = 0, 1, ...$, repeat

1. ADMM step (prediction step). Obtain $(\widetilde{\lambda}^k, \widetilde{\mu}^k, \widetilde{\nu}^k, \widetilde{a}^k, \widetilde{\phi}^k, \widetilde{\varphi}^k)$ in the forward order:

1.1 λ-minimization: Each front-end server i solves the following sub-problem for $\widetilde{\lambda}_i^k$:

$$\begin{aligned} \min \quad & -wU(\lambda_i) + \sum_j \left(\varphi_{ij}^k \lambda_{ij} + \frac{\rho}{2}(\lambda_{ij}^2 - 2a_{ij}^k \lambda_{ij}) \right) \\ \text{s.t.} \quad & \sum_j \lambda_{ij} = A_i, \lambda_{ij} \geq 0. \end{aligned} \tag{17}$$

This per-front-end server sub-problem (17) is of a much smaller scale compared to the original problem, with only N variables coupled by an equality constraint. We further observe that the objective function of the problem (17) described above is still convex, so problem (17) can be efficiently solved with the standard convex optimization technique.

1.2 μ-minimization: Each datacenter j solves the following sub-problem for $\widetilde{\mu}_j^k$:

$$\begin{aligned} \min \quad & (\phi_j^k + p_0)\mu_j + \frac{\rho}{2}(\alpha_j + \beta_j \sum_i a_{ij}^k - \mu_j - \nu_j^k)^2 \\ \text{s.t.} \quad & 0 \leq \mu_j \leq \mu_j^{\max}, \end{aligned} \tag{18}$$

where ϕ_j is the dual variable for the power balance constraint (15). This per-datacenter sub-problem (18) is a very easy quadratic program with only one variable, and it leads to:

$$\widetilde{\mu}_j^k = \max\left\{ \min\left\{ \alpha_j + \beta_j \sum_i a_{ij}^k - \nu_j^k - \frac{\phi_j^k + p_0}{\rho}, \mu_j^{\max} \right\}, 0 \right\}.$$

1.3 ν-minimization: Each datacenter j solves the following sub-problem for $\widetilde{\nu}_j^k$:

$$\begin{aligned} \min \quad & V_j(C_j \nu_j) + (p_j + \phi_j^k)\nu_j \\ & + \frac{\rho}{2}\big(\alpha_j + \beta_j \sum_i a_{ij}^k - \widetilde{\mu}_j^k - \nu_j\big)^2 \\ \text{s.t.} \quad & \nu_j \geq 0. \end{aligned} \tag{19}$$

This per-datacenter sub-problem (19) is with only one variable, thus, given the form of V_j, it can be readily solved. For instance, if V_j is a quadratic function or linear function, then the objective function is quadratic.

1.4 a-minimization: Each datacenter j solves the following sub-problem for $\widetilde{a}_j^k = (\widetilde{a}_{1j}^k, \cdots, \widetilde{a}_{Mj}^k)^T$:

$$\begin{aligned} \min \quad & -\sum_i a_{ij}(\beta_j \phi_j^k + \varphi_{ij}^k) + \frac{\rho}{2}\big(\beta_j \sum_i a_{ij}\big)^2 \\ & + \rho \sum_i a_{ij}(0.5a_{ij} - \widetilde{\lambda}_{ij}^k + \alpha_j \beta_j - \widetilde{\mu}_j^k \beta_j - \widetilde{\nu}_j^k \beta_j) \\ \text{s.t.} \quad & \sum_i a_{ij} \leq S_j, a_{ij} \geq 0. \end{aligned} \tag{20}$$

This per-datacenter sub-problem (20) is a quadratic program which has been studied in depth in the literature of convex optimization. Though it is large-scale, it can be transformed into a second order cone program and solved efficiently [13].

1.5 Dual update: Each datacenter j updates ϕ_j for the power balance constraint $\alpha_j + \beta_j \sum_{i \in \mathcal{S}} a_{ij} - \mu_j - \nu_j = 0$:

$$\widetilde{\phi}_j^k = \phi_j^k - \rho(\alpha_j + \beta_j \sum_i \widetilde{a}_{ij}^k - \widetilde{\mu}_j^k - \widetilde{\nu}_j^k).$$

Each front-end server i updates φ_{ij} for the equality constraint $a_{ij} = \lambda_{ij}$:

$$\widetilde{\varphi}_{ij}^k = \varphi_{ij}^k - \rho(\widetilde{a}_{ij}^k - \widetilde{\lambda}_{ij}^k).$$

2. Gaussian back substitution step (correction step). Obtain $(\lambda^{k+1}, \mu^{k+1}, \nu^{k+1}, a^{k+1}, \phi^{k+1}, \varphi^{k+1})$ in the backward order:

$$\begin{aligned} \phi_j^{k+1} &= \phi_j^k + \varepsilon(\widetilde{\phi}_j^k - \phi_j^k), \forall j, \\ \varphi_{ij}^{k+1} &= \varphi_{ij}^k + \varepsilon(\widetilde{\varphi}_{ij}^k - \varphi_j^k), \forall i, \forall j, \\ a_{ij}^{k+1} &= a_{ij}^k + \varepsilon(\widetilde{a}_{ij}^k - a_{ij}^k), \forall i, j, \\ \nu_j^{k+1} &= \nu_j^k + \varepsilon(\widetilde{\nu}_j^k - \nu_j^k) + \beta_j \sum_i (a_{ij}^{k+1} - a_{ij}^k), \forall j, \\ \mu_j^{k+1} &= \mu_j^k + \varepsilon(\widetilde{\mu}_j^k - \mu_j^k) - (\nu_j^{k+1} - \nu_j^k) \\ &\quad + \beta_j \sum_i (a_{ij}^{k+1} - a_{ij}^k), \forall j, \\ \lambda_{ij}^{k+1} &= \widetilde{\lambda}_{ij}^k, \forall i, j. \end{aligned}$$

Note that the Gaussian back substitution step is fairly simple in our problem, though it seems complicated in (10) with a complex matrix G. Furthermore, the computation cost of Gaussian back substitution step is very small, and is neglectable in comparison with the ADMM step.

IV. Performance Evaluation

In this section, we conduct trace-driven simulations to realistically evaluate the benefit of using fuel cell generation with intelligent control in geo-distributed datacenters. Our simulations are based on real-world workload traces, electricity price data sets, and electricity generation data sets.

A. Simulation Setup

We simulate a geo-distributed cloud that deploys $N = 4$ datacenters in four locations of North America: Calgary, San Jose, Dallas and Pittsburgh. Each datacenter's capacity is set to be uniformly distributed between $[1.7, 2.3] \times 10^4$ servers. We also assume that the cloud deploys $M = 10$ front-end proxy servers that are uniformly scattered across the continental United States.

Real workload trace: We use the one-week hourly HP request trace reported in the recent literature [23] to represent the request traffic of an interactive cloud service. The trace is scaled up proportionally and normalized to the number of servers required, as shown in the top of Fig. 3. We can see that the workload shows great variability and exhibits a clear diurnal pattern, which is typical for interactive cloud

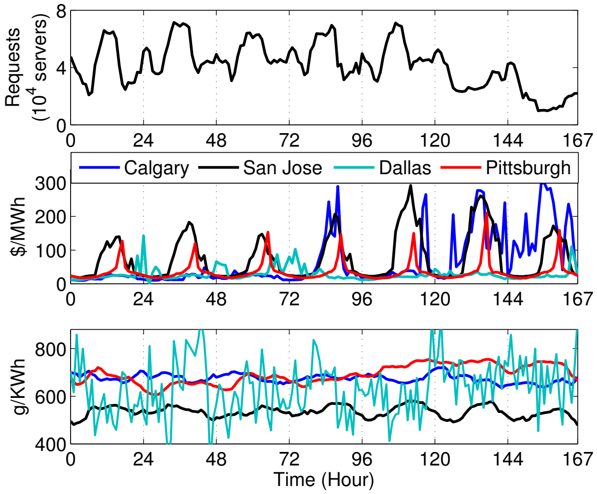

Fig. 3: The workload trace, electricity price data and carbon emission rate data at each datacenter.

service workloads. In order to imitate the geographical request distribution, we split this total workload among the $M = 10$ front-end proxy servers by following a normal distribution as in [16].

Electricity pricing data: We download the hourly locational marginal prices (LMP, in unit of \$/MWh) in real-time electricity markets of the four locations from the website of each regional transmission organization (RTO) or independent system operator (ISO). The time period of this data is September 10-16, 2012, including one week or 168 hours. This data is plotted at the middle of Fig. 3.

Electricity generation data: To estimate the carbon emission rate of each location hosting the datacenter, we first download the electricity generation data of each location from the website of the corresponding RTO or ISO. They report the hourly electricity fuel mix of each region for the major types of fuel. Then, we calculate the hourly carbon emission rates (in unit of g/KWh) of the four locations according to Eq. (1) given in Sec. II-B2, and plot it at the bottom of Fig. 3. The time period of this data is also September 10-16, 2012, the same as that of the electricity price data.

System Parameters: We first choose a higher energy efficiency level, *i.e.,* $\text{PUE}_j = 1.2$, for all of the four datacenters in our simulations. For the power consumption of the servers in each datacenter, we adopt a state-of-the-art setting. Specifically, each server has a peak power $P^{\text{peak}} = 200W$, and consumes $P^{\text{idle}} = 100W$ when idle. We assume that all four datacenters can be completely powered by fuel cell generation. Though difficult to meet at the current stage, it allows us to assess the most significant benefit that can be brought by fuel cells. Thus, we have $\mu_j^{\max} = P^{\text{peak}} \cdot S_j \cdot \text{PUE}_j, \forall j \in \mathcal{D}$.

We take the widely adopted quadratic utility function (2) to capture the user utility with weight $w = 10\$/\text{s}^2$ which makes the value of the user utility close to that of the electricity cost. We calculate the propagation latency L_{ij} according to the empirical approximation $L_{ij} = 0.02\text{ms/Km} \times d_{ij}$. The geographical distance d_{ij} can be obtained via mapping applications such as Google Maps. The emission cost function is the simple affine carbon tax function with a tax rate of $r_j = \$25/\text{ton}$, implying that emitting one ton of carbon will be charged \$25. Besides, the cost of fuel cell generation is set to be $p_0 = 80\$/\text{MWh}$, which is the state-of-the-art price. Finally, we set the penalty parameter $\rho = 0.3$ in our simulations.

B. Performance

To assess the benefits of using fuel cells with intelligent control, referred to as the *Hybrid* strategy, we further evaluate two alternative strategies: (1) *Grid*, which only uses electricity from the power grid and thus optimizes problem (12) with additional constraints $\mu_j = 0, \forall j$; and (2) *Fuel cell*, which only uses fuel generation and thus optimizes problem (12) with additional constraints $\nu_j = 0, \forall j$. We use I_{hg} to denote the cloud's utility improvement with the *Hybrid* strategy over the *Grid* strategy, use I_{hf} to denote the cloud's utility improvement with the *Hybrid* strategy over the *Fuel cell* strategy, and use I_{fg} to denote the cloud's utility improvement with the *Fuel cell* strategy over the *Grid* strategy.

Utility improvements. Fig. 4 depicts the utility improvements of the cloud under various strategies at each time period. We make the following observations. (1) Compared to the conventional strategy *Grid*, the emerging naive strategy *Fuel cell* would incur a UFC reduction up to 150% during electricity off-peak hours. During both load and electricity peak hours, the UFC improvement brought by *Fuel cell* strategy does not exceed 30%. (2) Further, the intelligent coordination between fuel cells and power grid is able to substantially improve the UFC, and the improvement is more than 40% on average when compared with *Fuel cell*. (3) Finally, using fuel cell generation with intelligent control is promising compared to depending only on the power grid, since it never reduces the UFC, yet it can bring prominent UFC improvements that can be up to 50% during electricity peak hours.

Insight: Entirely relying on fuel cell generation may incur a severe utility decline, while intelligently coordinating fuel cell generation with power grid can lead to a substantial utility improvement.

Load following capability. Fig. 5 plots the detailed workload performance in terms of average propagation latency under various strategies. Clearly, it demonstrates that completely relying on fuel cell generation will achieve the best workload performance, with average propagation latency between 14-16 ms, since requests are always routed to nearby datacenters. While for *Grid* strategy, routing requests to the datacenters with green and cheap energy will extend the workload propagation latency up to 23 ms. Finally, for *Hybrid* strategy, excellent workload performance is also achieved, with the average propagation latency between 14-17 ms, which is very close to that of *Fuel cell* strategy and far below that of *Grid* strategy.

Insight: Fuel cell generation enables load following capability described in Sec. II-B3 to greatly reduce the workload propagation latency.

Reduction of energy costs. Fig. 6 further compares the energy costs of various strategies. It can be seen that the *Fuel cell* strategy incurs the highest energy cost, since the price of fuel cell generation is more expensive than that of the electricity grid at most time periods. *This observation*

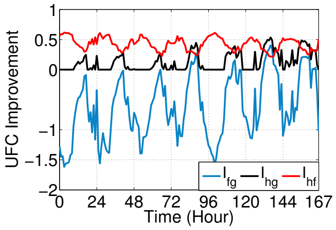

Fig. 4: UFC improvement under various strategies.

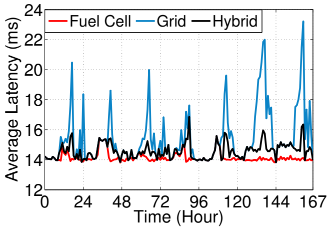

Fig. 5: Average workload propagation latency under various strategies.

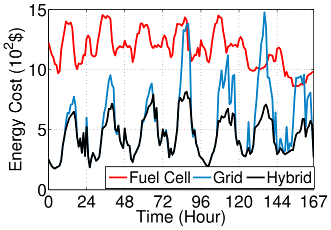

Fig. 6: Energy cost under various strategies.

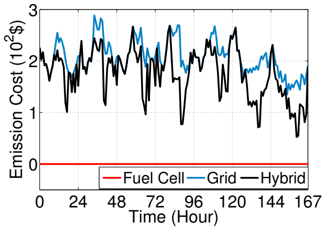

Fig. 7: Carbon emission cost under various strategies.

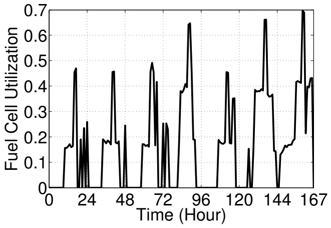

Fig. 8: The fuel cell utilization at each time period.

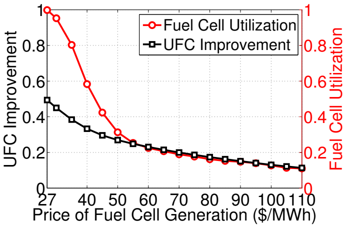

Fig. 9: The average UFC improvement and fuel cell utilization under various fuel cell generation price.

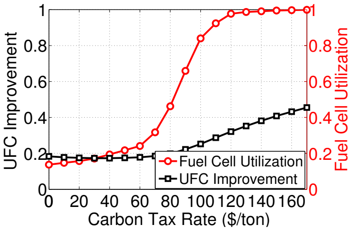

Fig. 10: The average UFC improvement and fuel cell utilization under various carbon tax rate.

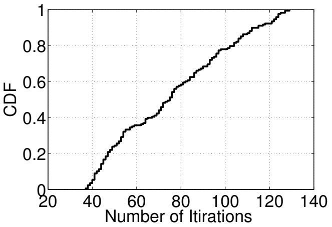

Fig. 11: CDF of the number of iterations to achieve convergence.

denies the reports that fuel cell generation is more economical than grid power [24]. However, such a massive cost can be substantially reduced by exploiting the price difference with the *Hybrid* strategy. In particular, Fig. 6 shows that such arbitrage can bring an energy cost reduction of almost 60%. Moreover, we also observe that compared to the *Grid* strategy, the *Hybrid* strategy can effectively reduce the energy cost during electricity peak hours, while it chooses to rely on the power grid at electricity off-peak hours as shown in Fig. 6.

Reduction of carbon emissions. We further investigate the carbon emission of different strategies. As we observe from Fig. 7, though using fuel cells can produce carbon-free generation, the *Hybrid* strategy, leveraging fuel cells with intelligent control, still incurs a significant amount of carbon emission, sufficiently close to that of the *Grid* strategy using only fossil fuels. Further, when compared to Fig. 6, we find that the energy cost dramatically outperforms the carbon emission cost, as the carbon tax rate ($25/ton) is lower than the electricity price. In theory, a lower carbon tax would put a stronger emphasis on the minimization of energy cost, and result in the frequent use of grid power during off-peak hours.

Fluctuations in fuel cell utilization. As implied by Fig. 6 and Fig. 7, the fuel cell generation was not frequently and massively used over 168 hours. For a more in-depth illustration, in Fig. 8, we plot the utilization of fuel cell generation in terms of the ratio of fuel cell generation to the power demand at each time period. We can clearly observe that the utilization of fuel cell generation fluctuates wildly, in order to respond to the bursty workload and electricity price, as well as time varying carbon emission rates. However, we also observe that the fuel cell generation is poorly utilized with a calculated average utilization at 16.2%, while the utilization never reaches 70%. This is because grid power is more competitive under the current high fuel cell generation price and low carbon tax rate.

Insight: The current high fuel cell generation price and low carbon tax rate make fuel cells poorly utilized in datacenters.

How low the price of fuel cell generation should go? The poor utilization of fuel cell generation shown in Fig. 8 motivates us to further investigate how low the price of fuel cell generation should go to promote the use of fuel cell generation in datacenters. Fig. 9 plots the average UFC improvement and fuel cell utilization for different values of the fuel cell generation price p_0 ($/MWh). We observe that, as the price of fuel cell generation decreases, both the UFC improvement and fuel cell utilization increase dramatically, and such improvements start to accelerate with the excessive decrease of p_0. Further, the figure shows that the current level of p_0 (80-110$/MWh) [15] leads to both poor UFC improvements ($11\% - 17\%$) and a lower fuel cell utilization ($11\% - 16\%$), while when p_0 decreases to 27$/MWh, the fuel cell utilization will reach 100%.

Does the carbon tax work well? We now investigate the impact of the carbon tax on promoting the use of fuel cell generation in datacenters. In particular, Fig. 10 plots the average UFC improvement and fuel cell utilization for different values of the carbon tax rate r ($/ton). It can be seen that, with the growth of the carbon tax rate, both the average UFC improvement and fuel cell utilization have improved remarkably. However, the average fuel cell utilization increases faster than the average UFC improvement, and converges to nearly 100% at a carbon tax rate of $140/ton. Further, the current level of carbon tax rates ($\$5 - 39$/ton) [1] has failed to promote either the fuel cell utilization or the UFC improvement (by more than 20%).

Convergence of our algorithm. We finally examine the convergence of our distributed ADM-G algorithm. Specifically, Fig. 11 plots the CDF of the number of iterations that our algorithm takes to achieve convergence for the 168 runs. It clearly shows that our algorithm is able to converge within 100 iterations for 80% of the total runs. Furthermore, the fastest run uses only 37 iterations, and our algorithm takes at most 130 iterations to converge, which remarkably outperforms some gradient or projection based methods that are reported to take hundreds of iterations to converge [25]. This demonstrates the fast convergence of our ADM-G based algorithm.

V. Related Work

The huge energy consumption and carbon emission caused by datacenters around the world have motivated extensive efforts on both intermittent renewable energy provisioning and cost/carbon-aware workload scheduling. For example, Li *et al.* [6] have explored the optimal allocation of solar power on multi-core systems, in order to improve the job performance and to maximize the use of solar energy. Goiri *et al.* [26] have investigated the benefits of dynamic interactive/batch workload scheduling and renewable/grid energy portfolio. Recent work [1], [7], [19], [20] have also explored load balancing across multiple datacenters to exploit the geographical diversity of electricity cost, carbon footprint or renewable availability. While recognizing their significance, our study is different from and complementary to the existing works. We take advantage of both the fuel cells' tunable output capacity and the grid power's geographically diverse carbon footprint to quantitatively examine the potential benefit brought by fuel cell generation, which has not been studied in the literature of cloud computing.

The ADMM technique that we use to maximize UFC has been widely used in machine learning and image processing. It was first brought to the field of cloud computing by Xu *et al.* [16] for workload management problems, where the classical ADMM is generalized to perform temperature aware workload management and capacity allocation for geo-distributed cloud services. However, the convergence of generalized ADMM depends on the assumption of strong convexity of the objective function, while in our UFC maximization problem, the carbon emission cost function may not be strongly convex. Thus, we introduce auxiliary variables to ADM-G (a different variant of ADMM proposed by He *et al.* [14], without relying on the strong convexity assumption) to develop a fully distributed algorithm.

VI. Concluding Remarks

In this paper, we present perhaps the first study on the fuel cell generation in geo-distributed cloud services. We argue through in-depth analysis and evaluation that fuel cell generation can be best utilized by an intelligent coordination between grid power and fuel cell generation. Specifically, we propose UFC, a quantitative index called the *utility of the cloud using fuel cells*, which captures the degree of the datacenters operator's satisfaction from the energy cost, carbon emission and workload performance. We formulate the UFC maximization problem as a general convex optimization, which jointly optimize the fuel cell generation and geographical request routing. To efficiently solve the large scale optimization problem, we propose a distributed algorithm by introducing auxiliary variables to decompose the global problem into a series of small scale problems, rather than directly applying the ADMM technique. Extensive trace-driven evaluations demonstrate that the use of fuel cell generation with intelligent control in geo-distributed datacenters leads to a significant improvement.

References

[1] P. X. Gao, A. R. Curtis, B. Wang, and S. Keshav, "It's Not Easy Being Green," in *Proc. of ACM SIGCOMM*, 2012.

[2] Google Green. [Online]. Available: http://www.google.com/green/bigpicture/#/intro/infographics-1

[3] The Guardian. [Online]. Available: http://www.guardian.co.uk/environment/2011/sep/08/google-carbon-footprint

[4] MIT Technology Review. [Online]. Available: http://www.technologyreview.com/view/507626.

[5] Datacenter Dynamics. [Online]. Available: http://www.datacenterdynamics.com/focus/archive/2013/04.

[6] C. Li, W. Zhang, C. Cho, and T. Li, "SolarCore: Solar Energy Driven Multi-core Architecture Power Management," in *Proc. of HPCA*, 2011.

[7] Y. Zhang, Y. Wang, and X. Wang, "GreenWare: Greening Cloud-Scale Data Centers to Maximize the Use of Renewable Energy," in *Proc. of Middleware*, 2011.

[8] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz, "The case for evaluating mapreduce performance using workload suites," in *Proc. of IEEE MASCOTS*, 2011.

[9] A. Qureshi, "Power-demand routing in massive geo-distributed systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2010.

[10] Carbon Usage Effectiveness (CUE): A Green Grid Data Center Sustainability Metric. [Online]. Available: http://www.thegreengrid.org/~/media/WhitePapers/CarbonUsageEffectivenessWhitePaper20101202.ashx.

[11] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010.

[12] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[13] H. Xu, C. Feng, and B. Li, "Temperature Aware Workload Management in Geo-distributed Datacenters," in *Proc. of USENIX ICAC*, 2013.

[14] B. He, M. Tao, and X. Yuan, "Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 313–340, 2012.

[15] Bloomenergy. [Online]. Available: http://www.bloomenergy.com/.

[16] H. Xu and B. Li, "Joint Request Mapping and Response Routing for Geo-distributed Cloud Services," in *Proc. of IEEE INFOCOM*, 2013.

[17] Z. Zhou, F. Liu, H. Jin, B. Li, B. Li, and H. Jiang, "On Arbitrating the Power-Performance Tradeoff in SaaS Clouds," in *Proc. of IEEE INFOCOM*, 2013.

[18] D. Niu, H. Xu, B. Li, and S. Zhao, "Quality-assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," in *Proc. of IEEE INFOCOM*, 2012.

[19] J. Yao, X. Liu, W. He, and A. Rahman, "Dynamic Control of Electricity Cost with Power Demand Smoothing and Peak Shaving for Distributed Internet Data Centers," in *Proc. of IEEE ICDCS*, 2012.

[20] Z. Zhou, F. Liu, Y. Xu, R. Zou, H. Xu, J. C. Lui, and H. Jin, "Carbon-aware Load Balancing for Geo-distributed Cloud Services," in *Proc. of IEEE MASCOTS*, 2013.

[21] C. Li, R. Zhou, and T. Li, "Enabling Distributed Generation Powered Sustainable High-Performance Data Center," in *Proc. of HPCA*, 2013.

[22] Power, Polution and the Internet. [Online]. Available: http://www.nytimes.com/2012/09/23/technology.html.

[23] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew, "Geographical Load Balancing with Renewables," in *Proc. of ACM GreenMetrics*, 2011.

[24] Doing The Math On Bloom Energy. [Online]. Available: http://www.forbes.com/2010/02/25/fuel-cell-costs-technology-ecotech-bloom-energy.html.

[25] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew, "Greening Geographic Load Balancing," in *Proc. of ACM Sigmetrics*, 2011.

[26] I. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy," in *Proc. of ACM ASPLOS*, 2013.