Holistic Management of Sustainable Geo-Distributed Data Centers

Zahra Abbasi*
Ericsson Research, San Jose, CA
zahra.abbasi@ericsson.com

Sandeep K.S. Gupta
IMPACT Lab, Arizona St. Univ., Tempe, AZ
sandeep.gupta@asu.edu

***Abstract*—This paper designs a holistic global workload management solution which explores diversities of a set of geo-distributed data centers and energy buffering in order to minimize the electricity cost, reduce the peak power drawn from utilities while maintaining the carbon capping requirement of the data centers. The prior work often designed solutions to address each of the aforementioned energy and cost optimization separately, disregarding the possible conflicts between the solutions' objectives. We propose a holistic solution to concurrently optimize the aforementioned potentially competing objectives. The proposed solution combines the techniques from Lyapunov optimization and predictive solution in order to manage the tradeoffs of electricity cost and carbon footprint reduction, and electricity cost and peak power cost reduction, respectively. The predicted data center parameters, being a significant aid to near optimally manage energy buffering and smoothing data centers' peak power draw, adversely affect the peak power cost due to the parameters' prediction error. The proposed holistic solution adapts stochastic programing to take the predicted parameters' randomness into consideration for minimizing the harmful impact of the prediction error. Our trace-based study confirms our analytical result that our holistic solution balances all the tradeoffs towards achieving energy and cost sustainability. Also our solution removes up to 66% of the prediction error impact in increasing the cost.**

***Keywords*—data centers, cloud computing, peak power, prediction error, carbon capping, electricity cost.**

I. Introduction

Internet data centers, typically distributed across the world in order to provide timely and reliable Internet service, have been increasingly pressurized to reduce their carbon footprint and electricity usage. Particularly, data centers will soon be required to abide by carbon capping polices which limit their maximum carbon footprint emission to encourage brown energy conservation [1]. Huge monthly electricity bill and costly power infrastructure of these data centers are other big concerns to the operators. Data centers spend 10 to 25 dollar per watt in provisioning their power infrastructure, regardless of the Watts actually consumed [2]. Since peak power needs arise rarely, provisioning power infrastructure for them can be expensive. Further, some utilities penalize data centers for their peak power consumption in addition to charging for the energy consumed (i.e., \$/W). Energy buffering management using energy storage devices (ESDs), e.g., existing UPSes, has been shown to be promising to shave the power demand, allowing aggressive under-provisioning of the power infrastructure [2]–[4]. Global workload management, i.e., intelligently distributing the workload across data centers according to their electricity price (\$/J) and carbon footprint (CO_2/J) at a given time, can also be of significant aid to shave the peak power drawn without requiring large-scale ESDs. Although energy buffering and global workload management have been considerably studied in the literature [5]–[9], the solutions designed so far are piecemeal in the sense that each of which addresses only some aspects of the problem. In particular, prior work has independently considered (i) electricity cost minimization and carbon footprint capping through an intelligent global workload management, and (ii) peak power cost reduction via energy buffering. We argue that there is a need for a holistic approach, which combines all the available leverages and concurrently optimizes the potentially conflicting objectives. Accordingly, we propose a new holistic global workload management for large-scale Internet services running in geo-distributed data centers. Such a holistic management, however, introduces new challenges.

First, the optimal solutions for the peak power cost minimization, energy buffering management and carbon capping can be only found offline. Prior online algorithms are designed to manage each (or two) of the aforementioned objectives separately, disregarding their implications on each other. Particularly, a window based predictive scheme, efficient for online management of peak power shaving [3], fails to competitively manage carbon capping with respect to the offline solution. This is because adjusting the carbon cap for each prediction window is difficult considering the intermittent nature of the available renewable energy. We use a combination of window based predictive scheme and T-slot Lyapunov optimization to jointly manage the energy cost (electricity and peak power cost) and the carbon footprint. The idea is to leverage the variability of data center parameters within the time frame T (e.g., a day) in order to smoothen the peak power drawn, and utilize Lyapunov technique to adjust the desired carbon footprint for each time frame over the entire budgeting period (e.g., a year). We hypothesize that such a solution achieves near optimal cost saving given the limited capacity of energy storage devices, and the daily variability of data center parameters.

Nevertheless, the efficiency of the previously discussed solution depends on the prediction accuracy of data center parameters over T, e.g., workload, electricity prices, and the available renewable energy. In particular, the prediction error has a very harmful impact on the peak power cost. The reason is that the optimal approach is to utilize the data centers with low electricity cost as much as possible without

This research has been funded by NSF CNS grant #1218505.
*The work was done when this author was with Arizona State University.

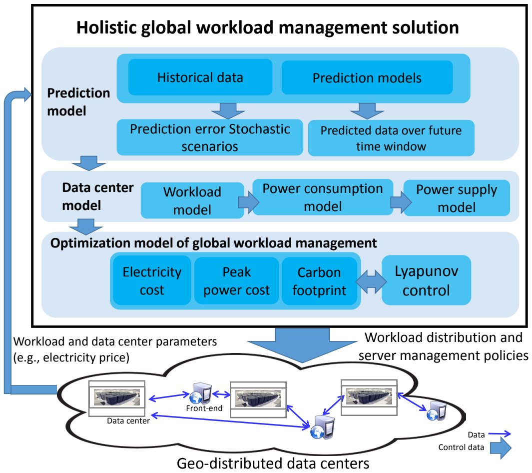


Fig. 1. Holistic global workload management solution.

increasing their peak power. Under any under prediction of those data centers' workload, for instance, their peak power most likely increases, resulting in an unexpected increase in the peak power cost. Previous prediction based schemes of peak power management are performed without considering the impact of the prediction error [4], [10]. We propose to make use of stochastic programming approach in order to mitigate the sensitivity of the solution to the prediction error. Accordingly, we formalize the cost minimization problem taking into account the stochastic scenarios, each of which represents the future realization of the cloud random parameters (e.g., input workload). A huge number of scenarios are required to completely describe the stochastic nature of the uncertainties associated with the cloud input parameters. Solving the stochastic cost minimization problem with those huge set of scenarios is computationally too expensive. So an appropriate scenario reduction technique must be used to limit the number of scenarios. We make use of the previously proposed scenario reduction algorithms [11] along with some problem-specific heuristics to reduce the number of scenarios. We hypothesize that a stochastic approach using a small set of stochastic scenarios, achieves a comparable performance against the solution with accurate data over T.

In summary we make the following contributions. We design a holistic global workload management solution which combines the potential benefits from a set of geo-distributed data centers to concurrently optimize the electricity cost, the carbon footprint and the peak power cost of data centers. As shown in Fig. 1, the holistic solution arranges and employs diverse set of models and techniques including predictive solution, stochastic programming and Lyapunov optimization to tackle energy management tradeoffs and enable coordination management of the energy cost and the carbon footprint. Throughout the paper, we incrementally enhance the holistic solution. We first frame the holistic global workload management problem as a linear programming (Section III), and develop a predictive solution, Online Cost minimization and Carbon Capping (OnCMCC), which utilizes T future slots' information. Next, we design a predictive Lyapunov based solution (OnCMCCLyp), which uses T-slot Lyapunov optimization technique to jointly minimize the cost and the carbon footprint across data centers (Section IV-C). OnCMCCLyp is proven to operate near offline optimal solution when T is sufficiently large and the predicted data are accurately available (Theorem 1). Next, we present our concluding holistic solution which adapts stochastic programming to model and solve the online solutions, in the presence of the parameters' prediction error (Section V). Finally, we perform a real-world trace based study to complement our analysis (Section VI).

II. Related work

There have been related efforts on reducing electricity bill and carbon footprint of data centers through workload management for a single [4]–[6], [8], [9] and a set of geo-distributed data centers [1], [7], [12]. In particular, related work proposed includes [1] and Lyapunov based optimization [9], [12], [13] for joint optimization of the electricity cost and the carbon capping. The efficiency of Lyapunov based solutions heavily depends on the value of the Lyapunov control parameter. There are also some recent works which explored the use of existing UPSes or any ESDs to reduce both the energy cost and the peak power cost for a single [2], [4]–[6], [8] and a set of geo-distributed data centers [10] without considering the carbon capping requirements of data centers. The related work, thereby, lacks a holistic solution to jointly manage the energy cost, the peak power cost and the carbon capping, a key solution for today's data centers to operate under carbon capping policies. This is important since such a holistic solution introduces new challenges which need to be addressed. Existing studies mainly used offline and predictive solutions for energy buffering management in data centers [4], [6], [8]. However, Lyapunov technique is used to exploit batteries in data centers for energy cost minimization [5]. The performance of the solution in [5] is based on restricting the maximum value of the Lyapunov control parameter, and the minimum required ESD capacity (which is relatively a large value). However, first, we seek a practical solution without requiring large scale ESDs to avoid their space and financial overhead. Second, the proposed solution only accounts for the energy cost. However ESDs can be best utilized to shave the peak power drawn, where its online management is shown to be effective when using window based predictive approach [3]. Third, using Lyapunov optimization for online management of both the carbon footprint and the ESD dynamics becomes a tedious task (if possible at all) since it requires a Lyapunov control parameter adjustment that optimally manages the two.

Further, the existing solutions on data center peak power shaving rely on the predictability of data centers' parameters over a window of time [2]–[4], [10], lacking analysis/solution to overcome the harmful impact of the prediction error on the peak power shaving. [14] designed an algorithm for single data centers to utilize Diesel Generators in order to compensate the impact of the prediction error in increasing the peak power cost. We use stochastic programming approach, to incorporate the randomness of the predicted parameters into the decision making process. Stochastic programming has been successfully applied in many applications, particularly, in grid power management and renewable energy optimization [11], [15]. However, we are the first (to our knowledge) to apply it for data center energy and power cost optimization.

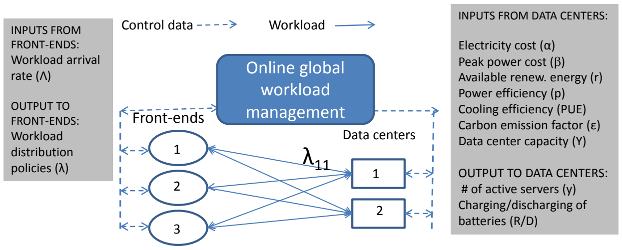


Fig. 2. System model.

TABLE I. Symbols and definitions.

Sym.	Definition	Sym.	Definition
t	slot index	ε^g	grid carbon emission
S	total # of slots	ε^r	renew. carbon emission
T	time frame ($T << S$)	Ψ	carbon cap
j	frontend index	$\psi=\frac{\Psi}{S}$	time-avg carbon cap
i	data center index	α	electricity price (\$/J)
N	# of data centers	β	peak power cost (\$/W)
Y	total # of servers	E	ESD capacity
λ	data centers' workload	d	ESD discharge rate
Λ	frontends' workload	c	ESD charge rate
p	per server power cons	D	ESD max discharge rate
g	grid power	C	ESD max charging rate
r	renewable power	η	ESD energy inefficiency
X	virtual queue	V	Lyap. control parameter
S'	peak power billing period	b_{max}	per-slot max carbon
p_0	stipulated peak power	ϕ	ESD cost per charge/discharge
ζ	prediction error rand. var.		
W	set of stochastic scenarios	R	feasible set of y and λ

III. System model and problem formulation

We consider a cloud, which consists of N geographically distributed data centers (see Fig. 2), where data center i has Y_i servers. We assume servers have only two states: active and inactive. Data centers get their required power from a mix of grid, on-site solar and wind renewable energy sources, and Energy Storage Devices (ESDs). We assume ESDs can be charged either from the grid or the available renewables.

We assume Internet workload for data centers, which typically exhibit daily and weekly seasonality and require timely services. End users' requests arrive from M geographically distributed front-ends (i.e., the sources), as shown in Fig. 2. The geographical front-ends may be network prefixes, or geographic groupings (states and cities). The workload management system operates in slotted time i.e., $t = 0 \ldots S-1$ for the budgeting period of S slots where the time slot length matches the timescale at which the server provisioning and energy storage charging/discharging cycle can be updated. In this paper we aim to design an online holistic global workload management which decides on the workload distribution and power management polices over every T slots, where $T << S$.

In the following sections, we incrementally design and enhance our holistic global workload management solution to conclude our final solution as shown in Fig. 1. Given models to describe the workload, the power demand and the power supply, we first frame the global workload management as an optimization problem. We argue that the joint management of the electricity cost, the peak power cost and the carbon footprint requires a combined technique of predictive solution and Lyapuonv optimization (Section IV-C). The resulting solution, however, is only effective when the parameters can be accurately predicted (zero prediction error). Given non-zero prediction error of the parameters, we design our final solution by taking into consideration the randomness of the predicted data (Section V), following the roadmap of the holistic solution given in Fig. 1.

IV. Optimization problem formulation

Our optimization problem uses models characterizing the power demand, derived from the workload distribution model, and the power consumption model of servers, and the power supply model which consists of models to describe the power drawn from the grid, batteries and the on-site renewables. The formulation and models build on the models used by the related work e.g., [4], [12]. The key change we make to [12] is to combine models of energy storage devices, peak power cost, and carbon capping in order to design a holistic solution for cost management and carbon capping of data centers.

Data Center Power Demand: Let $\Lambda_j(t)$ denote the average workload arrival rate at front-end j, our algorithm decides on the workload distribution of front-end j to data center i, denoted by $\lambda_{i,j}(t)$, and the number of active servers at each data center i, denoted by $y_i(t)$, subject to a set of workload requirements (e.g., delay requirement, availability of computation data) and data centers capacity. To model these requirements, we assume that $\lambda_\mathbf{i}(\mathbf{t}) = (\lambda_\mathbf{i,1}(\mathbf{t}), \ldots \lambda_\mathbf{i,j}(\mathbf{t}), \ldots, \lambda_\mathbf{i,M}(\mathbf{t}))$, and the corresponding $y_i(t)$ must be drawn from a feasible set, $(\lambda_\mathbf{i}(\mathbf{t}), \mathbf{y_i}(\mathbf{t})) \in \mathbf{R_i^{M+1}}(\mathbf{t})$. Our analysis works for any convex set of $R_i^{M+1}(t)$, which contains the constraints that $\sum_i \lambda_{i,j}(t) = \Lambda_j(t)$ (workload processing requirement), and $y_i(t) \leq Y_i$ (upper bound of number of servers). $R_i^{M+1}(t)$, for instance, can also contain the constraint that $\lambda_{i,j}(t)=0$ to represent the constraint that workload arriving at front-ends j cannot be processed at data center i due to the network latency or data availability. To solve our problem in Section VI, we use models in [1] which account for finding λ_i, and y_i based on M/M/n queuing model of data centers and an additive slack for number of severs to deal with workload spikes.

Given $R_i^{M+1}(t)$, the average one-slot energy consumption of an active server, denoted by p_i, can be obtained by profiling. Then p_i^{tot}, where, $p_i^{tot}=y_{i,t}p_i$ estimates the total one-slot energy consumed by active servers in data center i.

Data Center Power Supply: Data centers get their primary power from the grid. We perform our study under the dynamic pricing managed by the wholesale electricity market (e.g., north America). Under this model, the electricity pricing is dynamic, significantly varies over time and has seasonal daily, and monthly pattern. In addition to the energy actually consumed, some utility providers penalize the excess power draw: imposing additional fee if the peak power draw over a certain time window e.g., average power every 15 minutes [4], seen in a billing period (denoted by S') exceeds the stipulated power (denoted by p_0). Hence, we consider that the electricity price from the grid includes $\alpha_i(t)$, the usage price, and β_i, the surcharge per excess power draw over S' from p_0.

To model energy storage, we denote the energy storage level at time t by $e_i(t)$, and the charge/discharge energy during time slot t by $c_i(t)$ and $d_i(t)$, respectively. There is a limit on the maximum charging and discharging rate denoted by

C and D, respectively. An ESD has limited capacity, further it is associated with a cycle-life i.e., the average number of charging/discharging cycles in the lifetime of the device for a given depth of discharge. Furthermore, data centers reserver some of ESDs' capacity for use during the power outages. Therefore, we denote E the capacity of the ESD which can be used to manage the energy cost and the renewable energy utilization without affecting the data center availability and without violating the given depth of discharge. We assume that the efficiencies of ESD charging and discharging are the same, denoted by $\eta \in [0, 1]$, e.g., $\eta = 0.8$ means that only 80% of the charged or discharged energy is useful. Energy level of an ESD over time satisfies the following:

$$\begin{aligned}
\forall i,t: &\quad e_i(t+1) = e_i(t) + \eta_i c_i(t) - \tfrac{1}{\eta_i} d_i(t) \text{[ESD energy level]},\\
\forall i,t: &\quad 0 \leq e_i(t+1) \leq E_i, 0 \leq e_i(0) \leq E_i,\\
\forall i,t: &\quad 0 \leq c_i \leq C_i, 0 \leq d_i \leq D_i.
\end{aligned} \tag{1}$$

ESDs have some other physical limitations such as self discharge rate, which are ignored for notation brevity. Finally, in any slot, one can either recharge or discharge the battery or do neither, but not both. Hence, for all t and i we have:

$$\forall i,t: \quad c_i(t) d_i(t) = 0. \tag{2}$$

Consistent with today's data centers, we assume data centers get their power partially from the available on-site renewable energy (wind and solar) denoted by $r_i(t) \leq R_i$. For every data center i and all time t the energy demand and supply should be balanced as follows:

$$\begin{aligned}
\forall i,t: &\quad g_i(t) + r_i(t) + d_i(t) = p_i^{tot}(t) + c_i(t),\\
\forall i,t: &\quad g_i(t) \geq 0.
\end{aligned} \tag{3}$$

Cloud's Carbon Cap: To incorporate the carbon capping, we consider that each data center is associated with carbon emission intensities for the power source from utility denoted by $\varepsilon_i^g(t)$ and its on-site renewable denoted by $\varepsilon_i^r(t)$ in unit of CO_2 g/J. The total carbon footprint of the cloud, within slot t can be written as follows: $b_i(t) = \varepsilon_i^g(t) g_i(t) + \varepsilon_i^r(t) r_i(t)$. The cloud desires to follow the long-term carbon capping target, denoted by Ψ, which is typically expressed for a year of operation of a data center.:

$$\frac{1}{S} \sum_{t=0}^{S-1} \sum_i b_i(t) \leq \frac{\Psi}{S} = \psi, \tag{4}$$

where ψ denotes the time-averaged carbon cap.

A. Operational Cost Minimization and Carbon Capping

We set renewable energy operational cost to zero to maximize their utilization. In addition to the energy cost and the peak power cost, we consider that the data centers' operational cost accounts for the cost per maximum charging and discharging denoted by $\phi_{i,C}$, and $\phi_{i,D}$, respectively which depends on the ESD characteristics (e.g., cycle-life). The time-averaged operational cost of data centers over S slots, can be written as the following optimization problem, namely **P1**:

$$\begin{aligned}
\text{minimize} &\quad \tfrac{1}{S}\Big(\textstyle\sum_{t=1}^{S} \sum_i g_i(t)\alpha_i(t) + \frac{c_i(t)}{C}\phi_{i,C} + \frac{d_i(t)}{D}\phi_{i,D}\\
&\quad + \textstyle\sum_{t'=0}^{S/S'-1} \max_{(t'-1)S' \leq \tau \leq t'S'-1} (g_i(\tau) - p_0)^+ \beta_i \Big),\\
\text{subject to} &\quad (\lambda_\mathbf{i}(\mathbf{t}), \mathbf{y_i}(\mathbf{t})) \in \mathbf{R_i^{M+1}}(\mathbf{t})\, (1),\, (3), \mathbf{and} (\mathbf{4}).
\end{aligned} \tag{5}$$

To simplify the problem **P1**, note that Internet data centers typically contain thousands of active servers. So, we can relax the integer constraint of number of active servers (y_i) and round the resulting solution with minimal increase in cost. Also observe that **P1** disregards the non-convex and non-linear constraint (2), however the following lemma asserts that the optimal solution to **P1** never chooses to simultaneously charge and discharge from ESDs. This is intuitively clear, because charging and discharging the ESD in the same slot incurs additional battery cost and energy cost due to the battery inefficiency. It is, thereby, beneficial to instead satisfy the demand from the grid or do either charging or discharging.

Lemma 1. *The optimal solution to **P1** for every data center i and time t always chooses $c_i(t) d_i(t) = 0$.*

Lemma 1 can be proved by construction, which is deleted due to the space limitation (see [16, Lemma 7.1.1]).

The problem **P1** as described above is a linear programing (given a linear model for R^{m+1}) which can be optimally solved using the existing linear programming solvers. However, the solutions of **P1** over time are *dependent* due to the several sources of coupling factors: (i) the peak power cost is calculated over every $S' \geq 1$ slots (5), as a result it couples the solutions over S', (ii) the ESDs' dynamics (1) and the carbon capping constraint (4) couples the solutions over time. In practice, the billing period (S') is typically a month, and the carbon cap is typically given over a year of operation of the data centers. This means that S is typically equals to the number of slots for a year. Therefore, in practice, it becomes impractical to solve **P1** due to the unavailability of data as well as "curse of dimensionality". In this paper, we study and propose online solutions to solve **P1**. The performance of the online solutions are based on (i) the feasibility assumption which ensures that **P1** has non-zero feasible solutions, (ii) the bounded assumption which ensures that the total one-slot cloud's carbon footprint is bounded by b_{max}, i.e., $b(t) \leq b_{max}\ \forall t$, and (iii) the predictability assumption which ensures that the data center parameters are predictable over T slots with reasonable accuracy, and their most variabilities fall within T slots. Observe that, the assumptions are not constraining in practice, and that the last assumption is consistent with the daily variability of the data center parameters.

B. OnCMCC: predictive online solution

We design the online solution, namely OnCMCC, as a reference solution to solve the problem **P1** over $T \leq S'$, where T consists of slots for one or more days (e.g., T=24 or T=48 for hourly basis slots). In this solution we also use β' for the peak power cost where $\beta' = \frac{T}{S'}\beta$. OnCMCC is inspired by the observation that the variation of the data center parameters across days is usually lower than their variation across slots within days. Given the limited ESDs' sizes, the ESDs are most likely to be best utilized to leverage the daily variation of the data center parameters. The availability of renewable energy, however, not only significantly varies during days (solar energy is available only when sufficient sunshine is there), but also significantly varies during a days and even months in a year depending on the weather conditions and geographical locations. However, due to the limited size of ESDs and their physical limitations (e.g., self-discharge), it is impractical to migrate renewable energy across such long periods, making the cost optimality distance of OnCMCC negligible when carbon capping requirement is relaxed. Note, OnCMCC can only satisfy the carbon cap in a best-effort manner. Due to the intermittent nature of the renewable power, therefore, OnCMCC may significantly violate the carbon cap, making it inefficient particularly when cloud needs to perform under a relatively tight carbon capping requirement (i.e., ψ is comparable to that of the minimum carbon footprint possible). To avoid this problem, we extend OnCMCC to leverage the T-slot Lyapunov optimization in order to account for the dynamics of the carbon footprint.

C. OnCMCCLyp: T-slot Lyapunov based solution

In accordance with Lyapunov optimization, we define a virtual queue [17] with occupancy $X(t)$ equal to the maximum excess carbon footprint beyond the average carbon footprint over every T-slot. Using $X(0)$=0, we propagate the $X(t)$ values over every T-slot as follows:

$$X(t_0 + T) = max[X(t_0) - T\psi, 0] + \sum_{\tau=t_0}^{t_0+T-1} \sum_i b_i(\tau). \tag{6}$$

Building upon Lyapunov optimization technique we design OnCMCCLyp as given in Algorithm 1. The parameter V in Algorithm 1 is the Lyapunov control parameter which manages the electricity cost versus the carbon footprint reduction trade-off. OnCMCCLyp requires only T slots ahead information. The algorithm removes the coupling property of **P1** by (i) removing the constraint (4)), and (ii) managing the energy storage dynamics over window $(t, t{+}T{-}1)$ rather than S and managing peak power reduction over $(t, t{+}T{-}1)$, rather than S'. OnCMCCLyp uses T future slots information to manage the operational cost according to the variation of the parameters within the frame T and the Lyapunov technique to stabilize the carbon footprint dynamics across T-slots. In order to evaluate OnCMCCLyp, we theoretically compare its performance against the offline optimal solution of problem **P1** for the case of (i) S'=T, and (ii) the energy storage dynamics only depends on the window of T. In other words, we consider that the operational cost and energy storage can be optimally managed using T future slots information, and evaluate how OnCMCCLyp can manage the carbon cap (i.e., Ψ) without excessively increasing the operational cost.

Theorem 1. *(Performance Bound Analysis of OnCMCCLyp): Suppose* $X(0)$=0*, and that the maximum carbon footprint of the cloud over every T slot is upper bounded by Tb_{max}. Also define $cost_T^*$ as the optimal solution to the special case of problem* ***P1****, where S'=T, and for every t_0 the beginning slot in every frame T, we have $e_i(t_0 + T)+ = e_i(t_0)$. Further, suppose data center parameters are i.i.d. over every T-slots, and let cost(τ) and $b(\tau)$ denote the OnCMCCLyp cost and the carbon footprint, respectively for slot τ. Then for $V > 0$, and the integer variable $k = 0, 1, \ldots K$ where $S = KT$ we have the following:*

$$\begin{aligned} cost_T = \quad & \limsup_{K\to\infty} \tfrac{1}{K} \textstyle\sum_{k=0}^{K-1} \mathbb{E}\{\sum_{\tau=kT}^{kT+T-1} cost(\tau)\} \\ & \le cost_T^* + \tfrac{B}{V}, \end{aligned} \tag{7}$$

$$\sum_{t=0}^{S-1} \sum_i b_i(t) \le \Psi + \sqrt{2}\sqrt{KB{+}V(Kcost_T^*{-}\sum_{k=0}^{K-1}\sum_{\tau=kT}^{kT+T-1} cost(\tau)}), \tag{8}$$

where $B = \frac{1}{2}(T^2 b_{max}^2 + T^2\psi^2)$.

Algorithm 1 OnCMCCLyp Algorithm

Initialize the virtual queue X
for every k=1 $\ldots$ KT=S, where t=kT **do**
  Predict the system parameters over the window $t{+}T{-}1$
  Minimize:

$$\begin{aligned} & V\Big(\tfrac{1}{T}\textstyle\sum_{\tau=t}^{t+T-1}\sum_i g_i(t)\alpha_i(t) + \tfrac{c_i(t)}{C}\phi_{i,C} + \tfrac{d_i(t)}{D}\phi_{i,D} \\ & + \textstyle\sum_i \max_{t\le\tau\le t+T-1}(g_i(\tau) - p_{i,0})^+\beta_i\Big) - X(t)\textstyle\sum_{\tau=t}^{T+T-1}\sum_i b_i(\tau) \end{aligned} \tag{9}$$

  Subject to: $(\lambda_{\mathbf{i}}(\mathbf{t}), \mathbf{y_i}(\mathbf{t})) \in \mathbf{R_i^{M+1}}(\mathbf{t})$, (1), and (3).
  Update the virtual queue X using (6).
end for

Similar steps of [12, Theroem 1] can be taken to prove the Theorem.

From (8) and (7) it can be concluded that OnCMCCLyp achieves near optimal performance for sufficiently large value of S. According to Theorem 1, the OnCMCCLyp achieves average cost no more than a factor of $O(1/V)$ above the optimal average cost of P1 under the Theorem's conditions. The large value of V comes at the expense of an $O(V)$ tradeoff in achieving the carbon cap.

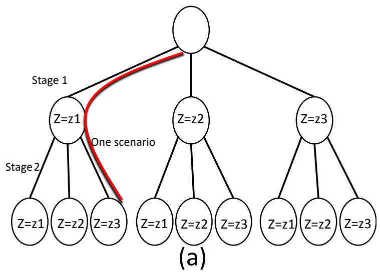
V. Stochastic Programming Approach

The performance of the online solutions depends on the predictability of the parameters over T. We use stochastic programming to take into consideration the randomness of the predicted input parameters. The major issue in developing the stochastic problem formulation is modeling of the uncertainties. We characterize and model uncertainties in the form of scenarios (possible outcomes of the data), a typical scheme in stochastic programming approach [18]. The goal is to find a policy that is feasible for all the possible parameter realizations (scenarios), and optimize the expectation of the objective functions given the probability associated with each scenario. Stochastic programming has many variants including stochastic dynamic programming. Stochastic dynamic programming in our problem requires discretization of the ESD states for every data centers in the cloud. This causes the number of states at each stage of the stochastic dynamic programming to dramatically increase. We, instead, choose to incorporate the stochastic scenarios in the original optimization problem and design the "deterministic equivalent" of the stochastic problem which is a typical stochastic programming approach [18]. Consider a deterministic optimization problem of the objective function f, the constraint function of h, and the decision variable of x, i.e., minimize $f(x)$, subject to $h(x)$. Its stochastic programming counterpart over set of stochastic scenarios of W can be written as follows:

$$\begin{array}{ll} \text{Minimize} & \sum_{w\in W} Pr(w) f(x, w), \\ \text{Subject to:} & h(x, w)\ \forall w \in W, \end{array} \tag{10}$$

where W denotes the set of scenarios, w denotes a scenario in W, and Pr denotes the probability function.

To characterize the scenarios, we model the prediction error of the input parameters, i.e., workload of each front-end, Λ_j, the available renewable power at each data center i, $r_i(t)$, the electricity price at data center location i, α_i, and the carbon intensity of the grid power at each data center i, ε_i^g. We denote $r_i(t)$ the actual renewable energy available to data center i at time t and use $\hat{r}_i(t)$ for the predicted generation. We denote



Fuel	CO_2/kWh
coal	986
PL	890
NG	440
NE	15
wind	22.5
solar	18

(b)

Fig. 3. (a) A sample scenario tree for the random variable Z over two stages, (b) carbon emission of electricity fuels (CO_2/kWh).

$r_i(t)=(1+\zeta_{i,r})\hat{r}_i(t)$, where $\zeta_{i,r}$ is the prediction error. We assume unbiased prediction error, i.e., $\mathbb{E}(\zeta_r)=0$, and denote the variance by σ_r^2 which can be obtained from historic data. These are standard assumptions in statistics. We use similar assumptions for the prediction error of the other input parameters. We also consider that the random variables (e.g., $\zeta_{j,\lambda}$) are independent random processes. As a result, the evolution of these stochastic processes is modeled as a multivariate random process. The marginal distribution for each of these random processes at any time step is assumed to be a normal distribution in accordance with the nature of unbiased prediction error. To define the scenarios, we approximate the marginal distribution of the random parameters (i.e., ζ) into discrete samples. The multivariate random process has therefore, $L_\lambda^M L_r^N L_\alpha^N L_\varepsilon^N$ samples at each time step, where L_λ, L_r, L_α, and L_ε denote the number of discretization levels used for the workload demand, the renewable power, the electricity price, and the grid carbon intensity, respectively. The evolution of the random process for the entire T slots is a huge set of scenarios. This type of uncertainty modeling results in a multistage "scenario tree" with T branching stages and $L_\lambda^M L_r^N L_\alpha^N L_\varepsilon^N$ samples at each node of the tree (see Fig. 3(a)). Each scenario (i.e., a path from root to a leaf of the tree) represents a possible future realization of the random process.

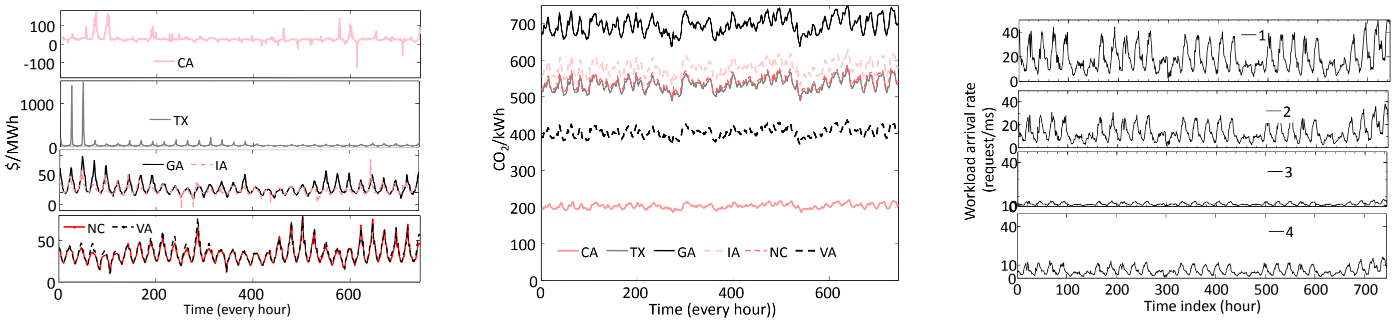
Observe that the scenario tree for our problem is huge. For instance for T=24, N=5, M=10, and L_λ=L_r=L_α=L_ε=5, the number of scenarios is 5^{600}. To solve the stochastic model, the multivariate random process with huge set of scenarios has to be approximated to a simple random process with small set of scenarios and should be as close as possible to the original scenario tree. There have been several scenario reduction algorithms in the literature which typically make use of probability metrics to choose a subset of scenarios [11], [19]. The scenario to be deleted is selected by comparing each scenario with the rest of the scenarios. Accordingly, the process of one to one comparisons of the scenarios needs to be repeated several times, which may not be feasible for a huge initial scenario tree. For instance, the scenario reduction algorithms in [11] make use of algorithms very similar to "k-means" and "k-medoids" where the probabilistic measures are used to evaluate the distance between the scenarios. Similar to k-means, these solutions can be implemented efficiently using parallel programming to run on a huge set of initial scenarios. As a general case, where running scenario reduction algorithms on the complete scenario tree may not be feasible, we can use problem-speccific strategies to generate a scenario tree with reasonable size as follows. Staregy one, **use stochastic aggregation rules to reduce the number of initial input random processes** (e.g., workload). Consider the random processes X and Y with normal distribution, $X \sim \mathbb{N}(\mu_1, \sigma_1^2)$, $Y \sim \mathbb{N}(\mu_2, \sigma_2^2)$, then $aX+bY$, where a and b are constant numbers, also has a normal distribution as follows, $aX+bY \sim \mathbb{N}(a\mu_1+b\mu_2, a^2\sigma_1^2+b^2\sigma_2^2)$. Data centers may use a combination of wind and solar energy where an aggregated random process of the two can capture their randomness. Also, in practice, number of front-ends (i.e., M) is very large. Suppose, every front-end can get service from all the available data centers in the cloud. Then, the entire input workload of all front-ends can be aggregated into one single random process. In practice, however, there are always some restrictions such as network latency (proximity of front-ends to the data centers) and data availability, where every front-ends can get service from a subset of data centers. In this case we can group front-ends depending on their feasible destination data centers and aggregate the workload of each group. Strategy two, **ignore random processes which have small standard deviation**. By removing such processes we significantly reduce the initial scenario tree size with negligible impact in the solution.

Following the model of (10), and given the set of scenarios W and the associated probability to each scenario $w \in W$, denoted by $Pr(w)$, we formulate the stochastic counterpart of the problem **P1**, namely **P2**. Next, we design our final holistic solution $\text{OnCMCCLyp}_{\text{stoch}}$, i.e., the stochastic counterpart of the online solution OnCMCCLyp based on **P2**. Following our road-map, given in Fig 1, $\text{OnCMCCLyp}_{\text{stoch}}$ combines leverages form data center and workload prediction models, stochastic programing, data center power consumption and power supply models, and Lyapunov optimization to concurrently optimize electricity cost, peak power cost and carbon footprint. Similar to $\text{OnCMCCLyp}_{\text{stoch}}$ the stochastic counterpart of OnCMCC, namely $\text{OnCMCC}_{\text{stoch}}$ is designed.

VI. Evaluation

We simulate a cloud consisting of six data centers located at CA, TX, GA, IA, NC and VA, most of which correspond to Google's data centers' locations, namely DC1, DC2, DC3, DC4, DC5, and DC6, respectively. The data centers are assumed to be homogeneous in terms of power consumption and computing characteristics, such that all the electricity cost savings and the carbon footprint reduction only comes from spatio-temporal variation of the electricity cost and the carbon footprint. Servers in each of the data centers are assumed to consume 300 W at peak utilization, and average response time and server slacks is set such that active servers have average utilization of 75% (active servers consume 250 W at this utilization). We set the slot length to one hour, S and S' to one month, and use realistic hourly traces of the electricity price (see Fig. 4(a) where data is taken from Locational Marginal Prices available at the corresponding RTO/ISO websites [1], and carbon intensity from Fig. 3(b) and fuel mix of data center locations. We also use the renewable traces of `http://www.nrel.gov/midc/` for three sites located in the data center locations of CA, TX and GA. To ensure data consistency, all traces are chosen from the month of July and August (see Fig. 5(a)). According to [4] a typical peak power cost is 12 $/KW per averaged power over 15-minutes slots.

[1] negative prices happen on the power wholesale market when a high power generation plant meets low demand.



(a) (b) (c)

Fig. 4. Hourly traces for one month : (a) electricity price (data taken from corresponding RTO/ISO website), (b) carbon emission, (c) workload (data taken from [20]).

Given our hourly basis slots we amortize β to 30 \$/KW. In most of the experiments we set p_0 as 80% of data centers maximum power consumption, unless stated.

We consider four front-ends, corresponding to four time-zones in the U.S., and use two months (July and August) of NASA workload Internet trace [20]. The workload of each front-end is scaled proportionally to the number of Internet users and shifted according to the time zone for each front-end in the corresponding area, as shown in Fig. 4(c). Each data center has 280 servers, and the intensity of the workload is such that at peak, 95% of servers in the entire cloud are required to be activated. We assume that a data center can receive workload from any of the front-ends.

We also consider a relatively large capacity for ESDs which can sustain the data centers for an hour. The results therefore, report a pessimistic performance of the online solutions, as large ESDs leverage the variabilities of prices and workload both within T slots and across T slots to minimize the cost. The physical characteristics of ESDs are set according to data sheet of Flood Lead Acidic (FLA) batteries used in data centers.

We use GNU Linear Programming Kit (GLPK) to solve problems **P1**, **P2** and the online algorithms.

Prediction results:: We use one month of training data (July traces) and build weekly and daily Seasonal Auto Regressive Integrated and Moving Average (SARIMA) prediction model (using "forecast" library of "R" package) to predict workload, and electricity prices and solar energy, respectively. Further, we use ARMA prediction model for wind energy. The lag one (one hour-ahead) prediction error is 14%, 20% and 25% for workload, electricity prices, solar and wind energy respectively. The error goes up to 20%, 40% and 54% for 24 lag (24 hour ahead) prediction of workload, solar and wind energy, respectively. Observe that the prediction error of both the solar and the wind energy in our data set is very high which can be typically improved using sufficient training data (using historical data of about 2-3 years [22]). Since, sufficient training data is not always available, we perform a pessimistic analysis on the impact of high prediction error on our solution, and the way that stochastic programming can remove its harmful impact. The prediction results of the electricity prices are very different across data centers. In particular, the electricity prices of DC4, DC5, and DC6 are predicted with relatively high accuracy, exhibiting error of 5% for lag one and 15% for lag 24. The electricity prices of DC1, DC2, and DC3, however, are predicted with low accuracy, exhibiting the error of 25% for lag one and 36% for lag 24. Due to the data insufficiency we do not build prediction model for carbon intensities and use accurate data.

Experiments performed: We perform experiments under different configurations: the length of T, the magnitude of the stipulated power, p_0, the magnitude of the carbon cap Ψ, and the prediction error. To evaluate OnCMCC and OnCMCCLyp, we use three reference solutions namely **Optimal** (optimal offline solution to **P1**), **MinCost** and **MinCarbon**. MinCost performs global workload management over the cloud to first minimize the cost and then the carbon footprint. **MinCarbon**, on the contrary, first minimizes the carbon footprint across the cloud and then the cost. MinCost and MinCarbon can be viewed as representative of the previous schemes which solely focus on either cost minimization (e.g., [7]) or carbon footprint minimization. We perform experiments to evaluate the incremental solutions of the global workload management scheme, which altogether framed the holistic solution. We first, evaluate OnCMCC to assess the efficiency of predictive solution for joint optimization of the electricity cost, and the peak power cost. Next, we evaluate OnCMCCLyp and compare it against OnCMCC to assess the combined predictive and Lyapunov based technique for coordinated management of the electricity cost, the peak power cost and the carbon footprint. Finally, we evaluate our final holistic solution OnCMCCLyp$_{\text{stoch}}$ to assess its performance for coordinated management of the electricity cost, the peak power cost and the carbon footprint in the presence of realistic predicted parameters and the prediction error.

A. *Joint optimization of cost and carbon capping*

We first relax the carbon capping constraint, and evaluate OnCMCC versus T and the magnitude of the stipulated power, p_0 (as percentage of data centers' maximum power). In order to run Optimal in a reasonable time, we run the experiments of this section using only three data centers (DC1, DC2, and DC3). The results, shown in Fig. 5(b), depicts that the larger the value of T, the closer the performance of OnCMCC becomes to that of Optimal. A daily basis T (T=24) can competitively manage the cost compared to Optimal even for large ESDs as long as p_0 is reasonably large. The magnitude of p_0 is typically such that such a power consumption arises

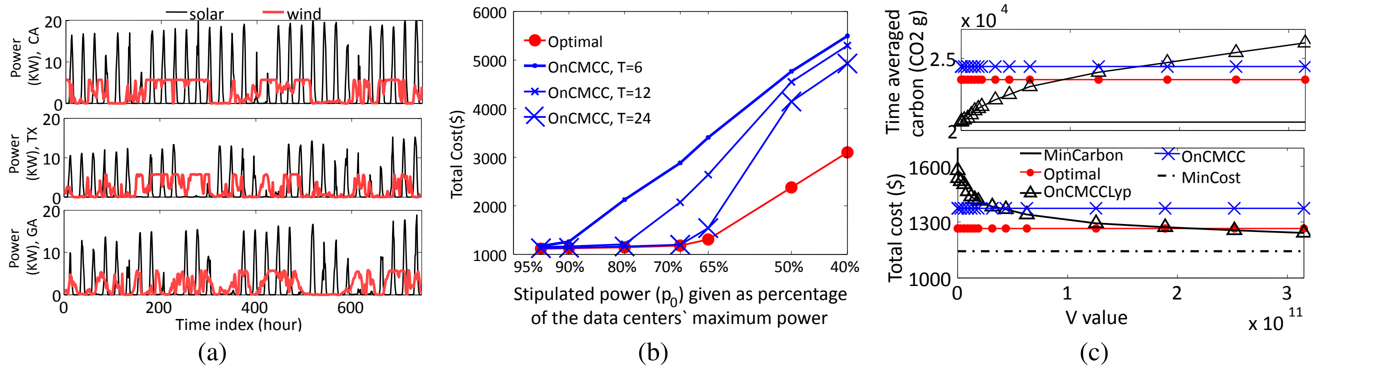


Fig. 5. (a) Hourly traces of solar and wind power (data taken from [21]), (b) total cost of OnCMCC versus stipulated peak power (p_0), and (c) performance of OnCMCCLyp versus Optimal and OnCMCC for various V values with tight Ψ.

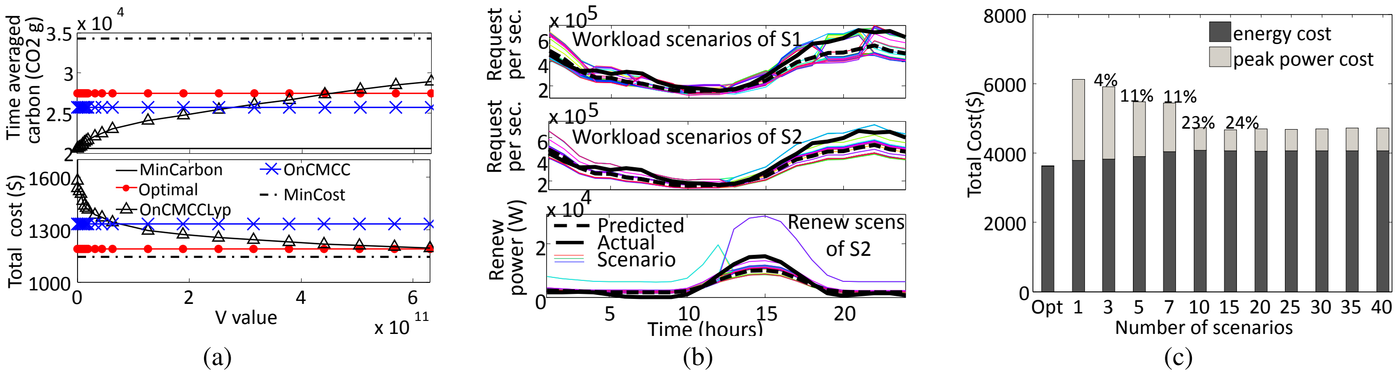


Fig. 6. (a) Performance of OnCMCCLyp for various V values and for Ψ is equal to the mean carbon footprint of MinCost and MinCarbon, (b) scenario tree of stochastic workload and renewable generation for a sample time frame T = 24, and (c) total cost of $\text{OnCMCC}_{\text{stoch}}$ versus $\text{OnCMCC}_{\text{opt}}$ (Opt) (the cost savings are calculated with respect to the one scenario case i.e., $\text{OnCMCC}_{\text{pred}}$) and number of S1 scenarios (zero renewable energy).

rarely, and a stipulated power which is 60% of the data center maximum power is an unrealistic value which is used to evaluate the worst-case performance of the solution.

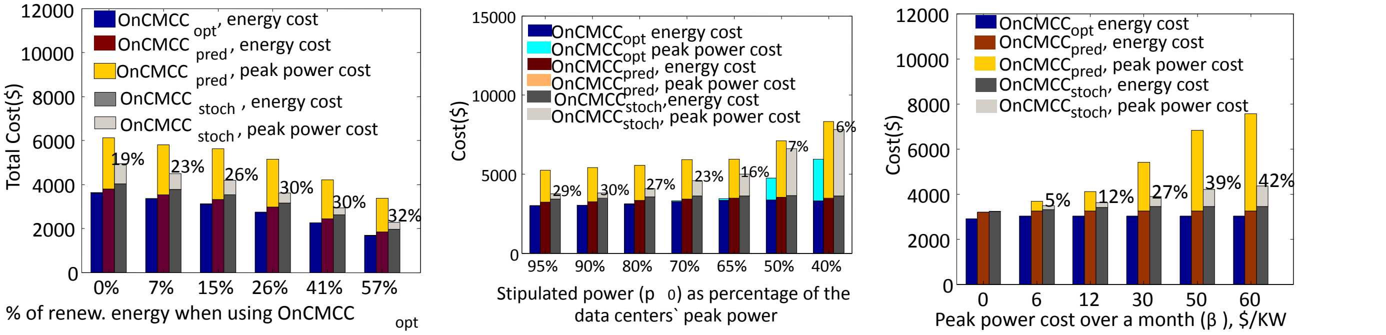
Next, we run MinCost and MinCarbon and perform some experiments to evaluate OnCMCC and OnCMCCLyp for T = 24, two values of the cap, Ψ, and various values of V, the Lyapunov control parameter. First, we set Ψ to a value very close to the carbon footprint achieved by MinCarbon. This is an example of the case where the cloud is associated with a tight cap. Results, shown in Fig. 5(c), depict that OnCMCC fails to meet the cap, whereas OnCMCCLyp meets the cap for V values less than 1.5×10^{11} (see Fig. 5(c)). Interestingly, Fig 5(c) shows that for V values in the range $[0.5\times10^{11}\ 1.5\times10^{11}]$, OnCMCCLyp yields lower carbon footprint and achieves lower energy cost (up to 7.5% lower cost) than that of OnCMCC. In particular, for a V value around 1.3×10^{11}, OnCMCCLyp performs very close to Optimal in terms of minimizing cost (sum of the electricity cost, the peak power cost and ESD cost) while satisfying the cap. Since OnCMCC independently manages the carbon footprint across T frames, it cannot opportunistically leverage the ups and downs of the cloud carbon footprint and the energy cost to optimally manage the two. OnCMCCLyp, however, takes the dynamics of the cloud carbon footprint into account and achieves a performance near to Optimal when V is appropriately adjusted.

Second, we set Ψ to the mean carbon footprint of MinCost and MinCarbon, example of the case where carbon cap is loose. Results, shown in Fig. 6(a), indicate that OnCMCC, in this case, achieves a lower carbon than that of Optimal, albeit at the expense of increasing the cost by 10%. OnCMCCLyp, however, for V values less than 4.5×10^{11} meets the cap. Similar to the previous case, OnCMCCLyp, when run with appropriate V value, outperforms than OnCMCC and achieves near Optimal performance in terms of minimizing the cost (see Fig. 6(a) for V values in the range $[2.5\times10^{11}\ 4.5\times10^{11}]$). In practice, OnCMCCLyp is expected to yield higher performance against OnCMCC when performed for more than one month, since the carbon intensity variations over months are huge.

Although the results of Theorem 1 is based on the assumption of T=S' (in the experiment S'=S=168 i.e., one month), the experimental results running for T=24<<S' show that OnCMCCLyp achieves near one competitive ratio against Optimal for an appropriate V value. From the above results we conclude that T-slot Lyapunov based solution, OnCMCCLyp, is indeed effective for using as a holistic solution to manage the electricity cost, the peak power cost and the carbon capping. Note, the appropriate V value depends on the cloud parameters e.g., the carbon footprint. The parameter B in Theorem 1, gives a clue for adjusting V. The results so far, however, are given for the case where the T slot future information are accurately available. Next section evaluates the solutions when using predicted data over T slots.

B. Stochastic optimization

We characterize $\zeta_{r,i}$, ζ_λ and $\zeta_{\alpha,i}$ through the prediction results. Then the marginal distribution of each of them, cover-



(a) (b) (c)

Fig. 7. Total cost of $\text{OnCMCC}_{\text{stoch}}$ versus $\text{OnCMCC}_{\text{opt}}$ (Opt) using 15 scenarios of S2: (a) various renewable energy utilization, (b) various stipulated power (p_0), and (c) various peak power cost (β).

ing 90% confidence interval, is approximated to five samples each with equal probabilities. Due to their large differences, parameters' samples are normalized between zero and one. We use $\zeta_{r,i}$ to represent the aggregated prediction error of both the wind and the solar energy at each data center and ζ_λ to represent the aggregated prediction error of the workload for all front-ends (see Section V). Given a random process at one stage, we construct the scenario tree over T and apply [11, Algorithm 2] to construct two reduced scenario sets: (i) S1 solely from the discrete marginal distribution of ζ_λ, and (ii) S2 from the discrete marginal distribution of ζ_λ, $\zeta_{r,i}$, and $\zeta_{\alpha,i}$. In order to run [11, Algorithm 2] in a reasonable time, we evolve the scenario trees of S1 over eight stages, and S2 over two stages. Fig. 6(b), shows that S1 and S2 capture the randomness of the predicted workload more accurately than that of the predicted renewable energy due to its high prediction error. We evaluate OnCMCC and OnCMCCLyp when using predicted data over T=24 (namely $\text{OnCMCC}_{\text{pred}}$ and $\text{OnCMCCLyp}_{\text{pred}}$) versus when using stochastic programming approach (namely $\text{OnCMCC}_{\text{stoch}}$ and $\text{OnCMCCLyp}_{\text{stoch}}$) and when using accurate data (namely $\text{OnCMCC}_{\text{opt}}$ and $\text{OnCMCCLyp}_{\text{opt}}$). We run stochastic solutions for different number of scenarios ($\text{OnCMCC}_{\text{stoch}}$ of one scenario is identical to $\text{OnCMCC}_{\text{pred}}$). In the figures we show the sum of the electricity cost and the battery cost as energy cost.

Number and type of scenarios: First, we set the renewable energy of all data centers to zero and use S1. From the results of Fig. 6(c), it can be seen that the prediction error has a harmful effect on the peak power cost. In particular, while $\text{OnCMCC}_{\text{opt}}$ can manage grid power draw to avoid the peak power cost, $\text{OnCMCC}_{\text{pred}}$ with one scenario incurs \$2400 for the peak power, increasing the total cost by 66% compared to $\text{OnCMCC}_{\text{opt}}$. The total cost of $\text{OnCMCC}_{\text{stoch}}$ is decreased from 6% for 3 scenarios up to 24% for 15 scenarios compared to the total cost of $\text{OnCMCC}_{\text{pred}}$ (i.e., $\text{OnCMCC}_{\text{stoch}}$ of one scenario). This means that $\text{OnCMCC}_{\text{stoch}}$ yielding \$900 more cost than $\text{OnCMCC}_{\text{opt}}$ (as opposed to \$2400 for $\text{OnCMCC}_{\text{pred}}$), can remove 62.5% of the harmful prediction error impact in increasing the cost. Hence, the results agree with our initial hypothesis that stochastic programming with small number of scenarios can significantly mitigate the harmful impact of the prediction error. Fig. 6(c) also shows that the peak power cost saving of $\text{OnCMCC}_{\text{stoch}}$ with multiple scenarios, compared to its deterministic counterpart ($\text{OnCMCC}_{\text{pred}}$), comes at the expense of a slight increase in the energy cost. Further, the performance of $\text{OnCMCC}_{\text{stoch}}$ does not improve when number of scenarios increases beyond 15. Note that stochastic programming does not guarantee an optimal performance, and its performance heavily depends on the problem, the predicted error magnitude, and the scenarios.

Next, we fix the number of scenarios of S2 to 15, and scale the renewable energy of DC1, DC2, and DC3 such that the total renewable energy utilization of the cloud varies from 0% to 57% when using $\text{OnCMCC}_{\text{opt}}$. Results, as shown in Fig. 7(a), similar to that of Fig. 6(c), indicates that $\text{OnCMCC}_{\text{stoch}}$ when using S2 significantly removes the impact of the prediction error of the workload, the electricity prices, and the renewables (removing 66% and 89% of the prediction error impact for 15% and 57% renewable energy utilization cases, respectively). The less scenario coverage of S2 for the predicted workload compared to that of S1, causes the performance of $\text{OnCMCC}_{\text{stoch}}$ to downgrade by 5% (compare 24% cost saving of $\text{OnCMCC}_{\text{stoch}}$ in Fig. 6(c) with 19% in Fig. 7(a) for the case of 0% renewable utilization). The cost saving of $\text{OnCMCC}_{\text{stoch}}$ increases compared to its deterministic counterpart ($\text{OnCMCC}_{\text{pred}}$) with increasing the availability of the renewable energy. This is because taking the randomness of the renewable and workload prediction error into consideration results in higher utilization of the renewable energy and consequently decreasing the cost. The impact of such a management is higher for the higher availability of the renewable energy.

We also evaluate the performance of $\text{OnCMCC}_{\text{stoch}}$ (when using S2 with 15 scenarios and 15% renewable energy utilization case) for various stipulated peak power (p_0) and peak power cost (β). Fig. 7(b) shows that the cost saving of $\text{OnCMCC}_{\text{stoch}}$ against $\text{OnCMCC}_{\text{pred}}$ is higher for higher stipulated power where stochastic scenarios can significantly affect the decisions. Fig. 7(c) indicates that the cost saving of $\text{OnCMCC}_{\text{stoch}}$ against $\text{OnCMCC}_{\text{pred}}$ is higher for higher β. Generally, $\text{OnCMCC}_{\text{stoch}}$ incurs very similar expected electricity cost to that of $\text{OnCMCC}_{\text{pred}}$, this is the reason that $\text{OnCMCC}_{\text{stoch}}$ has a total cost almost equal to that of $\text{OnCMCC}_{\text{pred}}$ for the case where $\beta = 0$. With increasing the peak power cost the impact of prediction error on increasing the peak power cost of $\text{OnCMCC}_{\text{pred}}$ is worsen which can be mitigated using $\text{OnCMCC}_{\text{stoch}}$.

Finally, we set the carbon cap, Ψ, to the mean carbon footprint of MinCost and MinCarbon and run $\text{OnCMCCLyp}_{\text{stoch}}$

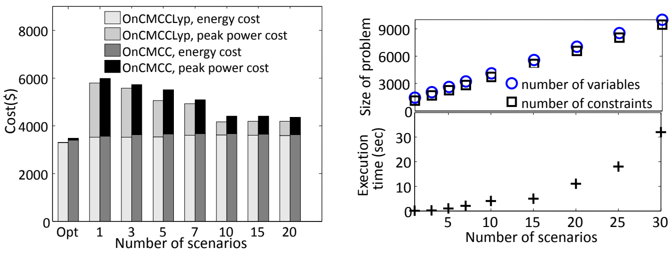


Fig. 8. Total cost of $\text{OnCMCCLyp}_{\text{stoch}}$ and $\text{OnCMCC}_{\text{stoch}}$ versus $\text{OnCMCC}_{\text{opt}}$ (Opt) and number of scenarios of S2, and (b) Overhead of $\text{OnCMCC}_{\text{stoch}}$.

with appropriate V value over different number of scenarios of S2. The results, as shown in Fig. 8(a), have a similar trend to those of the previous results (e.g., Fig. 6(c)) in the sense that the stochastic programming solutions ($\text{OnCMCC}_{\text{stoch}}$ and $\text{OnCMCCLyp}_{\text{stoch}}$), significantly remove the impact of the prediction error, improving the cost of $\text{OnCMCC}_{\text{pred}}$, and $\text{OnCMCCLyp}_{\text{pred}}$ up to 30% by using ten scenarios (removing the impact of the prediction error by 66%). This cost saving comes at a slightly energy cost increase as shown in Fig. 8(a) and consequently a slightly carbon footprint increase.

Overhead of the stochastic solution: The cost efficiency of the stochastic programming solutions comes at the expense of increasing the size of the optimization problems. As a result, the execution time of the solutions increases depending on the computing system's capability. Fig. 8(b) shows that the size of the optimization problem of OnCMCC_{stoch} linearly increases with increasing the number of scenarios in terms of both the number of decision variables and the number of constraints. This translates into the exponential increase in the execution time of the solution in our testbed (Intel Quad core i7-3770 CPU 3.4GHz, and 8G memory). Therefore it is important to run the stochastic solutions with small number of scenarios and an efficient implementation.

VII. Conclusions

We proposed a holistic global workload management solution, which jointly minimizes data centers operational cost (including peak power cost), while satisfying the carbon capping requirement of the geo-distributed data centers. Peak power cost management, energy buffering and carbon capping all introduce time coupling in the solution. We developed an online algorithm OnCMCCLyp which (i) leverages (daily) predictability of data center input parameters to efficiently manage energy storage dynamics and to smoothen the power draw from the grid, and (ii) uses T slot Lyapunov optimization to manage the cost carbon footprint tradeoff. Our trace based study shows that our T-slot Lyapunov based solution, OnCMCCLyp can achieve near one competitive ratio with respect to the optimal offline solution when the Lyapunov control parameter is appropriately adjusted, T is sufficiently large and data over T is accurately available. However, the prediction error of the parameters over T slots has a very harmful impact on the peak power shaving and consequently on the cost efficiency of the solution. Our proposed stochastic programming approach is shown to remove up to 66% of such negative impacts.

References

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

[2] S. Govindan, D. Wang, A. Sivasubramaniam, and B. Urgaonkar, "Aggressive datacenter power provisioning with batteries," *ACM Transactions on Computer Systems (TOCS)*, vol. 31, no. 1, p. 2, 2013.

[3] A. Bar-Noy, M. P. Johnson, and O. Liu, "Peak shaving through resource buffering," in *Approximation and Online Algorithms*. Springer, 2009, pp. 147–159.

[4] D. Wang, C. Ren, A. Sivasubramaniam, B. Urgaonkar, and H. Fathy, "Energy storage in datacenters: what, where, and how much?" *ACM SIGMETRICS Perf. Eval. Rev.*, vol. 40, no. 1, pp. 187–198, 2012.

[5] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramanian, "Optimal power cost management using stored energy in data centers," in *Proc. ACM SIGMETRICS*, 2011, pp. 221–232.

[6] S. Govindan, A. Sivasubramaniam, and B. Urgaonkar, "Benefits and limitations of tapping into stored energy for datacenters," in *ISCA*. IEEE, 2011, pp. 341–351.

[7] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," in *Proc. ACM SIGCOMM*, 2009, pp. 123–134.

[8] V. Kontorinis, L. E. Zhang, B. Aksanli, J. Sampson, H. Homayoun, E. Pettis, D. M. Tullsen, and T. S. Rosing, "Managing distributed UPS energy for effective power capping in data centers," in *ISCA*. IEEE, 2012, pp. 488–499.

[9] A. H. Mahmud and S. Ren, "Online capacity provisioning for carbon-neutral data center with demand-responsive electricity prices," *ACM SIGMETRICS Perf. Eval. Rev.*, vol. 41, no. 2, pp. 26–37, 2013.

[10] M. Etinski, M. Martonosi, K. Le, R. Bianchini, and T. D. Nguyen, "Optimizing the use of request distribution and stored energy for cost reduction in multi-site internet services," in *SustainIT*. IEEE, 2012.

[11] N. Growe-Kuska, H. Heitsch, and W. Romisch, "Scenario reduction and scenario tree construction for power management problems," in *Power Tech Conference Proceedings*, vol. 3. IEEE, 2003, pp. 7–pp.

[12] Z. Abbasi, M. Pore, and S. K. Gupta, "Online server and workload management for joint optimization of electricity cost and carbon footprint across data centers," in *IPDPS*. IEEE, May 2014.

[13] 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 *IEEE MASCOTS*, 2013.

[14] Z. Liu, A. Wierman, Y. Chen, B. Razon, and N. Chen, "Data center demand response: Avoiding the coincident peak via workload shifting and local generation," *Performance Evaluation*, vol. 70, no. 10, pp. 770–791, 2013.

[15] V. S. Pappala, I. Erlich, K. Rohrig, and J. Dobschinski, "A stochastic model for the optimal operation of a wind-thermal power system," *Power Systems, IEEE Transactions on*, vol. 24, no. 2, pp. 940–950, 2009.

[16] Z. Abbasi, "Sustainable cloud computing," Ph.D. dissertation, Arizona State University, 2014.

[17] M. J. Neely, "Energy optimal control for time-varying wireless networks," *Information Theory, IEEE Transactions on*, vol. 52, no. 7, pp. 2915–2934, 2006.

[18] A. Shapiro, D. Dentcheva *et al.*, *Lectures on stochastic programming: modeling and theory*. SIAM, 2009, vol. 9.

[19] M. Kaut and S. W. Wallace, "Evaluation of scenario-generation methods for stochastic programming," *Pacific Journal of Optimization*, vol. 3, no. 2, pp. 257–271, 2007.

[20] M. F. Arlitt and C. L. Williamson, "Web server workload characterization: The search for invariants," in *ACM SIGMETRICS Perf. Eval. Rev.*, vol. 24, no. 1, 1996, pp. 126–137.

[21] [Online]. Available: http://www.nrel.gov/midc/

[22] M. G. De Giorgi, A. Ficarella, and M. Tarantino, "Error analysis of short term wind power prediction models," *Applied Energy*, vol. 88, no. 4, pp. 1298–1311, 2011.