Cost Minimization using Renewable Cooling and Thermal Energy Storage in CDNs

Stephen Lee
School of Computer Science
UMass, Amherst

Rahul Urgaonkar
IBM Watson

Ramesh Sitaraman
School of Computer Science
UMass, Amherst

Prashant Shenoy
School of Computer Science
UMass, Amherst

***Abstract*—Content delivery networks employ hundreds of data centers that are distributed across various geographical locations. These data centers consume a significant amount of energy to power and cool their servers. This paper investigates the joint effectiveness of using two new cooling technologies - open air cooling (OAC) and thermal energy storage (TES) - in CDNs to reduce their dependence on traditional chiller-based cooling and minimize its energy costs. Our Lyapunov-based online algorithm optimally distributes workload to data centers leveraging price and weather variations. We conduct a trace based simulation using weather data from NOAA and workload data from a global CDN. Our results show that CDNs can achieve at least 64% and 98% cooling energy savings during summer and winter respectively. Further, CDNs can significantly reduce their cooling energy footprint by switching to renewable open air cooling. We also empirically evaluate our approach and show that it performs optimally.**

I. INTRODUCTION

Modern content delivery networks (CDNs) allow content providers of web-based services to efficiently deliver their content to end-users through a global network of servers. From an architectural standpoint, servers of a CDN are organized into clusters that are housed in a distributed set of data centers across the globe. Given their immense sizes, distributed data centers of a CDN consume significant amounts of energy, and as a result, the design of green CDNs that reduce the energy usage of this Internet-scale distributed system has gained recent research attention [1], [2].

CDNs, and data centers in general, consume significant amounts of energy to power their servers and cool them. While techniques for reducing the energy consumed by data center servers have received significant attention [3], [4], techniques for reducing the energy spent in cooling servers of a CDN are less well studied. The topic is nevertheless important since cooling energy represents a significant portion of the total energy usage within data centers—in some cases, up to a watt of cooling energy is needed for each watt consumed by the servers [9]. In an effort to reduce their cooling energy, companies such as Facebook have made remarkable progress by switching to renewable sources for their cooling energy needs. By using renewable cooling sources, rather than expensive chillers, they have significantly reduced the energy needed to cool their data center servers and achieved Power Usage Effectiveness (PUE) as low as 1.07 [5].

Based on these emerging trends, in this paper, we focus on using alternate technologies, including renewables, for cooling servers in distributed data centers of a CDN to achieve reductions in cooling energy usage as well as energy costs. We study two complementary cooling technologies: renewable open air cooling (OAC) and thermal energy storage (TES) to achieve these goals. In case of renewable OAC, outside air is directly used to cool the servers of the data center, instead of relying on traditional chiller-based techniques. Since open air cooling is largely "free", it can achieve significant reductions in energy cooling costs. However the effectiveness of the approach depends on outside weather conditions—the outside temperature and humidity must permit its use and it may not be feasible to employ OAC during extreme weather conditions such as very hot or humid summer days. Thermal energy storage (TES) is a complementary cooling technology where thermal energy is stored by the data center in chilled water or chilled ice tanks, and this stored energy is used to cool data center servers when needed—e.g., when OAC becomes infeasible. While TES has not seen much use in data center scenarios, they are common in other settings such as manufacturing plants [6]. The use of thermal energy storage as a failover option to OAC is similar to the use of UPS batteries—a form of chemical energy storage—as a failover option to power servers when grid power becomes unavailable. Thermal energy storage techniques can also be used to optimize cooling energy costs by storing energy when electricity prices are low and using stored thermal energy during peak hours when prices are high.

Thus, OAC and TES are complementary technologies and both have the potential to significantly reduce the reliance on expensive chiller-based cooling technologies in data centers and CDNs. A CDN has significant flexibility at its disposal in exploiting these technologies. Typically CDN content is replicated at multiple data centers for reasons of availability and performance. In such a scenario, the CDN routes user requests to the nearest data center that has the requested content to optimize user-perceived performance. In the presence of OAC and TES, the decision on which data centers to use for servicing user requests can be made based on *both* energy and performance considerations. When the weather permits the use of OAC, requests continue to be served by the nearest data center like before. When the weather does not permit the use of OAC at a particular CDN data center, the CDN can dynamically decide between one of two options: it can switch over to the use of TES at such sites or it can redirect request to another data center in the region where the local weather allows for the use of OAC. Further, if a data center is low on stored thermal energy and none of the nearby data centers can employ OAC, the CDN can also choose to redirect requests to nearby data centers that have available TES capacity. Clearly the decision on which CDN data center to use for incoming requests and whether to use OAC or TES must be made online and dynamically in an autonomous fashion. To do so, we need an online adaptive approach that determines how to best use OAC and TES at various CDN data centers while ensuring that users are serviced from the closest feasible data center for performance reasons.

Our contributions: In this paper, we propose a new Lyapunov optimization-based distributed algorithm that integrates the use of OAC and TES for cooling servers of a CDN. Our approach makes online decisions on how to adaptively route requests to maximize the use of OAC and intelligently uses TES when OAC is infeasible within a region; the algorithm also makes intelligent decisions on when to charge the TES and when to use the stored thermal energy based on variable electricity pricing schemes. While Lyapunov-based approaches have previously been proposed for using batteries to optimize server power usage [3], its use in thermal energy storage settings for cooling servers and the integration of OAC with TES has not been studied previously. We evaluate our algorithm using real workload traces from a large commercial CDN and real weather and electricity pricing data from various locations. Our results show that CDNs can achieve at least 64% and 98% cooling energy savings during summer and winter respectively (Figure 1). Further, we show that load redirection can be restricted to minimally impact performance. We also show that CDNs can significantly reduce its cooling energy footprint as 99% of the cooling is provided by OAC during winter (Figure 4). We provide theoretical bounds on our algorithm's performance (Theorem 4.1), and empirically evaluate to show that its performance is close to optimal (Figure 2).

II. Background

In this section, we present background on the architecture of a CDN as well as the cooling energy model and cooling technologies assumed in our work.

A. Content Delivery Network Architecture

A content delivery network (CDN) comprises thousands of servers, organized into a distributed network of data centers spread across geographical locations. A CDN can leverage its proximity to end users to deliver content while minimizing latency and loss. Architecturally, a CDN employs a two level load balancing algorithm that first maps incoming requests to a particular data center and then maps each request to a particular server within that data center. The first-level load balancer, which is referred to as the global load balancer, will typically choose a data center that is closest to the end-user to optimize user-perceived performance. Our work enhances the global load balancer to make it energy aware, specifically to take advantage of new cooling technologies when determining how to map requests to data centers, while continuing to optimize user performance.

B. Cooling energy model

To model server power consumption, we use a well-known linear model for computing the power consumed by a server [7]. The power consumed by a server (in Watts) can be modeled as a function of its workload i.e. $P_{idle} + (P_{peak} - P_{idle})\lambda$, where P_{idle} is the power consumed by an idle server, P_{peak} is the power consumed during peak load and λ is the normalized workload equivalent to the load served by the server as a fraction of its capacity. In order to calculate the server energy consumed in a data center, we further assume that the servers can be dynamically consolidated [8].

Given the energy consumed by a server using the above model, we can then derive a simple model to compute the energy needed to cool server. Basically, we transform the server energy to cooling energy using Power Usage Effectiveness (PUE), an energy efficiency metric for data centers, which is a ratio between the total energy consumed in a data center and its total server energy. Given the data center PUE and the sever energy consumption, we can calculate the cooling enegy.[1] Recent surveys have shown that the average PUE of a data center is 1.8 [9], which is the value assumed in our experiments when estimating cooling energy.

C. Cooling technologies

Open air cooling: Data centers can be equipped with OAC and can use one of the two OAC technologies: air-side or water-side economizer. In an air-side economizer, hot air is flushed outside the data center and cool air is drawn into the data center. Water-side economizer, on the other hand, uses water as a medium to cool, and uses cooler towers to enable "free cooling". Depending on existing infrastructure of the data center, either of the technologies can be integrated to avail "free" cooling. The American Society of Heating, Refrigeration, and Air conditioning Engineers (ASHRAE) defines the permissible temperature and humidity for operating IT equipments. We assume that whenever the outside dry-bulb temperature, dew-point temperature and humidity are below the ASHRAE recommended maximum, a data center at that location can rely on outside air to cool its server [10]. Modern server hardware have been built to withstand significantly higher temperature and humid levels. For example, an ASHRAE class 2 server is engineered to withstand temperatures of up to 35 C and humidity levels of 80%. More recent ASHRAE class 4 server hardware can withstand temperatures of up to 45 C and humidity levels of 90%, making it feasible to use OAC even in locations with warm or humid climates [10]. We assume class 2 data centers in our experiments.

Thermal Energy Storage: TES allows cooling energy to be stored in a medium (ice, water etc.) providing flexibility in drawing power from grid. Most facilities uses TES for load shifting. By storing energy when prices are cheap and discharging when prices are high, load is shifted in time to minimize the total energy cost. Our work assumes that TES is provisioned to meet peak demand of the data center.

Chillers: Most HVAC systems use water as a medium to cool the data center. Using chillers, heat is removed from the water and recycled back to HVAC. Since removing heat from liquid consume a lot of power, it is not surprising that chillers consume one-third power in the data center [11]. Thus, limiting the use of chiller can bring substantial cost benefits in a data center.

[1] PUE = Total Power / IT Power and Total Power = (IT Power + Cooling Power); Given the PUE and Server Power, we can compute cooling power.

III. Problem Statement: An Offline LP Formulation

We formulate the problem addressed in our work as an optimal offline linear program (LP). The linear program takes as input complete knowledge of the (future) workload at each data center, future electricity prices and weather data to compute which data center services what portion of the workload and how each data center is cooled while minimizing total cooling energy cost. Intuitively the approach uses "free" OAC locally when possible or redirects load to other nearby data centers where OAS is feasible if it is infeasible locally. When OAC is not possible locally or at other nearby locations, local (or remote) TES is used; chiller-based cooling may be used if it is cheaper or when TES capacity is depleted. Our optimal offline LP, while impractical in practice, provides a baseline for comparison with an online approach.

Note that we are interested in minimizing the sum total cooling power cost across multiple data centers by making use of the OAC and TES technologies in an optimal manner. This problem requires global decisions about routing of workloads across multiple data centers as well as local decisions about charging/discharging of TES and servicing the workload using a combination of OAC, TES and power drawn from the grid. Note that at each data center, the incoming workload, availability of OAC, and the electricity price vary over time, thereby making this a challenging control problem. The LP formulation we present below highlights the control decisions and constraints involved in this problem.

We begin by defining our model. We assume there are N data centers and that time is slotted. In each slot t, we denote the original workload intended for data center i by $\lambda_i(t)$. For each data center i, let K_i denote the set of data centers where this workload can be routed. As an example, this could be based on a distance metric that ensures that the resulting routing delay is tolerable. Note that $\forall i, i \in K_i$.

Next, the amount of workload routed from data center i to j is denoted by $\lambda_{ij}(t)$. These $\lambda_{ij}(t)$ must satisfy the following conservation constraint:

$$\sum_{j \in K_i} \lambda_{ij}(t) = \lambda_i(t) \;\; \forall i \tag{1}$$

Next, the total incoming workload to data center i (after routing) is given by $\sum_j \lambda_{ji}(t)$. We must have that

$$\sum_{j} \lambda_{ji}(t) \leq \mu_i \;\; \forall i \tag{2}$$

where μ_i is the capacity of data center i. This workload results in a cooling power demand according to the model described earlier and must be satisfied using a combination of OAC, TES and power drawn from the grid.

The availability of OAC at data center i in slot t is denoted by a $0/1$ variable $O_i(t)$. Since OAC use incurs no power cost, without loss of generality we assume that all available OAC is first used to satisfy the cooling power demand and that any remaining demand is served using TES and power drawn from the grid. Denote the remaining demand (after OAC) at data center i by $W_i(t)$. For simplicity, we assume that when $O_i(t) = 1$, then all cooling power demand can be satisfied using OAC alone. Thus, in this case, $W_i(t) = 0$. However, our formulation can easily be extended to consider the case where only part of the cooling power demand can be satisfied using OAC alone.

Next, we denote the TES recharge and discharge amounts at data center i by $R_i(t)$ and $D_i(t)$ respectively. Also denote the total power drawn from the grid by $P_i(t)$. Then the following equality must be satisfied for all i, t

$$W_i(t) = P_i(t) - R_i(t) + D_i(t) \;\; \forall i, t \tag{3}$$

We assume that the $R_i(t), D_i(t)$ and $P_i(t)$ are upper bounded by $R_i^{\max}, D_i^{\max}$ and $P_i^{\max}$ respectively.

Finally, the recharge and discharge decisions affect the stored energy in the TES as follows. Let $Y_i(t)$ denote the amount of stored energy in the TES of data center i in slot t. Denoting its efficiency by $0 < \alpha \leq 1$, we have:

$$Y_i(t+1) = Y_i(t) + \alpha R_i(t) - D_i(t) \;\; \forall i, t \tag{4}$$

We assume that the TES has a maximum capacity of $Y_i^{\max}$. Since the stored energy cannot be negative, we must have that

$$0 \leq Y_i(t) \leq Y_i^{\max} \;\; \forall i, t \tag{5}$$

Let $C_i(t)$ denote the unit electricity price at data center i in slot t. Also, let $C_i^{\min}, C_i^{\max}$ denote the minimum and maximum values taken by $C_i(t)$. Then the sum total power cost over an interval $[0, T]$ is given by

$$\sum_{t=1}^{T} \sum_{i=1}^{N} P_i(t) C_i(t) \tag{6}$$

The LP formulation seeks to minimize the objective in (6) subject to all the constraints discussed so far. This is a linear program since the objectives as well as all the constraints are linear. However, it should be noted that the complexity of this LP increases with both N and T and it becomes infeasible to solve it for large instances. Further, solving this LP requires full knowledge of the entire workloads, prices as well as OAC availability. In the next section, we present an online solution to this problem that overcomes both of these limitations.

IV. Online Algorithm

In this section, we present an online control algorithm for the power cost minimization problem presented in Sec. III. This algorithm is based on the technique of Lyapunov optimization [12] and is similar in spirit to the algorithm presented in [4] for the problem of power cost optimization in a single data center using batteries. Here, we extend this algorithm to consider multiple data centers along with the availability of OAC.

The online control algorithm operates as follows. First, we define a shifted version $X_i(t)$ of stored energy level $Y_i(t)$ for each data center i as follows

$$X_i(t) = Y_i(t) - \frac{V C_i^{\max}}{\theta_i} - D_i^{\max} \tag{7}$$

where the parameters V and θ_i are constants defined as

$$V = \min_i \Big(\frac{Y_i^{\max} - R_i^{\max} - D_i^{\max}}{C_i^{\max} - C_i^{\min}/\alpha} \Big) \quad (8)$$

$$\theta_i = V \Big(\frac{C_i^{\max} - C_i^{\min}/\alpha}{Y_i^{\max} - R_i^{\max} - D_i^{\max}} \Big) \quad (9)$$

and where we assume that $V \geq 0$. Given the collection of $X_i(t)$, the algorithm makes joint decisions about routing of workloads as well as charging/discharging TES by solving the following optimization problem every slot.

$$\max \sum_{i=1}^{N} D_i(t)\big(\theta_i X_i(t) + V C_i(t)\big) - V \sum_{i=1}^{N} W_i(t) C_i(t) + \sum_{i=1}^{N} R_i(t)\big(-\alpha\theta_i X_i(t) - V C_i(t)\big) \quad (10)$$

This optimization is subject to constraints (1), (2), (3) as well as the upper bounds on $R_i(t), D_i(t)$ and $P_i(t)$. It should be noted that this results in a much simpler LP compared to the offline formulation presented earlier. In fact, we can further simplify this LP by observing the following two structural properties of its optimal solution:

- If $\theta_i X_i(t) + V C_i(t) < 0$, then $D_i(t) = 0$, i.e., there is no discharge for TES of data center i in slot t.
- If $\alpha\theta_i X_i(t) + V C_i(t) > 0$, then $R_i(t) = 0$, i.e., there is no recharge for TES of data center i in slot t.

The above properties follow by noting that their corresponding terms in the objective are maximized by choosing the $D_i(t)/R_i(t)$ values to be 0. After implementing the output of this optimization, the online algorithm proceeds by updating the values of $X_i(t)$ and repeats this procedure.

We make the following observations about this algorithm. First, it is online, requiring no knowledge of future prices, workloads or OAC availability. Second, this algorithm is easy to implement, requiring solving a simple (and much smaller) LP in each slot. Further, we next show both theoretically and empirically that the cost achieved by our online technique is within a bounded additive term of the solution generated by the LP. This additive term can be made arbitrarily small by scaling the TES capacity. This is formally shown by the following theorem.

Theorem 4.1: Suppose the online algorithm given by (10) is implemented over T slots with a control parameter V as defined in (8). Then, the following hold:

1) Each queue $X_i(t)$ is deterministically upper and lower bounded for all t as follows:

$$-\frac{V C_i^{\max}}{\theta_i} - D_i^{\max} \leq X_i(t) \leq Y_i^{\max} - \frac{V C_i^{\max}}{\theta_i} - D_i^{\max} \quad (11)$$

2) The TES energy level $Y_i(t)$ satisfies for all i, t:

$$0 \leq Y_i(t) \leq Y_i^{\max} \quad (12)$$

3) Suppose the processes $C_i(t), O_i(t)$ and $\lambda_i(t)$ are i.i.d. over slots. Then the expected per slot cost under the online algorithm is within B/V of the optimal offline value. i.e.,

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\{P_i^{\text{online}}(t) C_i(t)\right\} \leq \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\left\{P_i^{\text{lp}} C_i(t)\right\} + \frac{\sum_i B_i}{V} \quad (13)$$

where B_i is a constant (independent of V) defined as

$$B_i = \frac{(D_i^{\max})^2 + (R_i^{\max})^2}{2} \quad (14)$$

Proof: The proof is based on the technique of Lyapunov optimization [12] and can be found in the Appendix. ■

The performance bound (13) shows that the gap between the offline LP cost and the online cost can be made small by increasing V.

V. EXPERIMENTAL EVALUATION

In this section, we first describe our experimental methodology and then present our results.

A. Experimental Methodology

We use a trace-based simulation to analyze the potential benefits using the Lyapunov-based algorithm discussed in Section IV. Our extensive traces contain a month-long workload traces collected from Akamai CDN, a year-long weather trace provided by National Oceanic and Atmospheric Administration (NOAA) and a year-long real-time pricing trace (RTP) shown in Table I. The workload trace contains load information, total capacity, number of servers deployed, location information such as city, latitude and longitude etc. The load information is captured at a granularity level of 5 minutes. On the whole, the trace contains 390 US locations, including a total of 63045 servers spread across all locations. For the purpose of our evaluation, we selected six US data center locations as our representative sample. Unless stated otherwise, all of our experiments use all six locations. The weather trace contains hourly dew-point, dry-bulb temperature, humidity, location information for the year 2012. Finally, our pricing data contains hourly pricing information for the year 2011 and 2013.

To determine OAC feasibility for all locations at a given time, we first identify the weather stations closest to these locations. Since, the location of data centers and the weather stations are known, we could map each data center to its closest weather station. Next, we use the dry-bulb temperature and dew point to determine whether the conditions are within the ASHRAE standards for OAC. Unlike the five minutes granularity of the workload information, the weather data is hourly. Hence, we assume the weather to remain the same for a given hour. This assumption is reasonably correct since the weather doesn't change rapidly over short time intervals. Given that we had only a month-long workload trace, we repeat our workload pattern for each month of the year and use the weather data to create a combined trace containing OAC feasibility information for each location for the entire year.

ISO	Locations	Duration
California ISO	San Jose	Jan - Dec, 2011
ERCOT ISO	Houston	Jan - Dec, 2013
Midcontinent ISO	Chicago	Jan - Dec, 2013
NEMassBOST	Boston	Jan - Dec, 2013
New York ISO	New York	Jan - Dec, 2013
PJM ISO	Philadelphia	Jan - Dec, 2013

TABLE I: Real-time pricing dataset

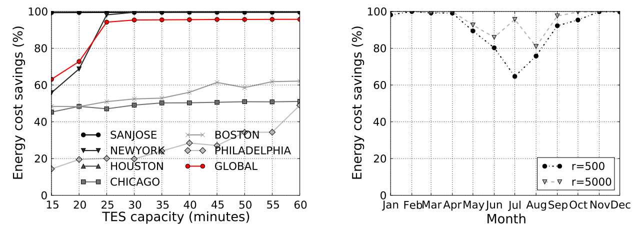
Next, we compute the server and cooling energy required in each data center using the combined trace containing OAC feasibility information. The approach to compute the server and energy is described in Section II. Finally, we compute the reduction in energy cost using our dynamic price history for various TES capacities and distances of radius r. The radius r parameter imposes a load redirection constraint on each data center, and restricts the movement of load to data centers that are within the radius r. Such restrictions on load movement is useful for requests that are sensitive to latency. For all our simulations, we assume TES starts with full capacity and its efficiency $\alpha = 1$.

B. Empirical Results

1) Cooling energy cost savings using OAC and TES: We analyze the ability of OAC and TES to minimize the total cooling energy cost. We ran our Lyapunov-based algorithm for the entire year with varying TES capacity and radius r. Note that the energy cost reduction is either from free OAC, or due to using TES, or by redirecting to cheaper price locations. Thus, to evaluate the potential energy cost savings, we compare it against a baseline where no OAC or TES cooling technology or any load redirection mechanism to cheaper electricity price location is available. For each data center and each time slot t, the algorithm decides whether to service the load locally or remotely, charging/discharging TES based on the workload, price, OAC feasibility, and TES capacity.

Since OAC is "scarce" during summer months, we plot the energy cost savings for the month of July with $r = 5000$ kms (see Figure 1 (a)). Note that an increase in TES capacity increases the overall energy cost savings. In particular, the overall energy cost savings increase from 63% to 95%. However, the cost savings see diminishing returns after TES capacity of 30 minutes. While increasing TES does not add additional cost savings, a large TES capacity allows peak-demand shaving and is useful in a peak-based pricing scheme. In addition, it can store renewable energy from intermittent sources such as solar or wind, reducing its dependency on brown energy. Interestingly, the individual cost savings of some cities may decrease with increase in TES capacity. However, the overall cost savings increases with increase in TES capacity. Specifically, the cost savings for Chicago decreases from 48% to 47% with increase in TES capacity from 20 to 25 minutes. Such a behavior is observed when servicing a load remotely is cheaper than servicing locally, decreasing the savings of the data center but increasing the overall savings.

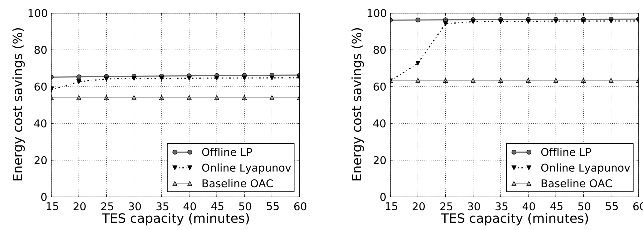
Figure 1 (b) shows the overall energy cost savings for all six US locations with radius $r = 5000$ and $r = 500$ kms and TES capacity of 45 minutes. Note that at least 98% and



(b) Cost savings in July. r = 5000 kms

(a) Annual cost savings. TES = 45 mins

Fig. 1: Energy cost savings across six US locations



(a) r = 500 kms

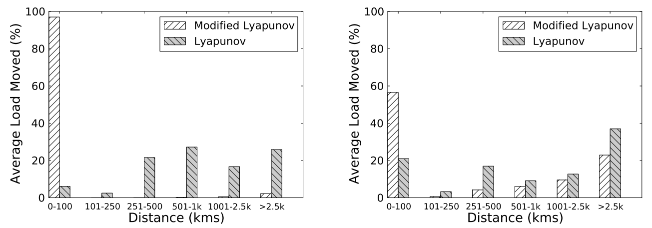
(b) r = 5000 kms

Fig. 2: Performance comparison between offline LP and online Lyapunov for the month of July

100% cost savings is achieved during winter months with $r = 500$ kms and $r = 5000$ respectively. Since OAC is "free", the additional savings in $r = 5000$ kms is achieved due to load redirected to a distant datacenter where OAC is feasible and cooling capacity is available.

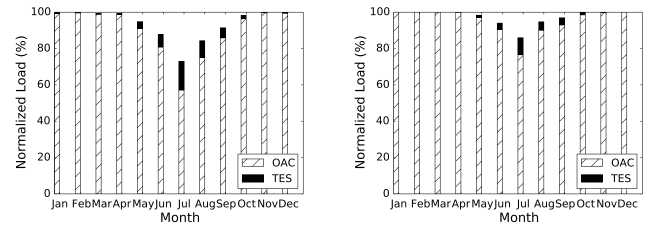
2) Convergence of our Lyapunov approach: We validate the convergence of our algorithm stated in Theorem 4.1 by comparing it against an offline LP described in Section III. To compare the offline LP with online Lyapunov algorithm, we compute the cost savings using both approaches. We also compare the cost savings against a baseline OAC – cost savings from using only OAC – and calculated using a greedy approach. In this greedy approach, load is redirected from no OAC available data centers to OAC available data centers. subject to the distance and capacity constraints i.e. load is redirected only to a data center is within a radius r and does not exceed the data center's cooling capacity. Figure 2 (a) and (b) shows the result for $r = 500$ and $r = 5000$ kms respectively. Note that the gap between the offline LP and online algorithm reduces with the increase in TES capacity. In fact, with TES capacity as little as 30 minutes our approach reaches close to the offline algorithm.

3) Load balancing and its impact on performance: To study our online algorithm's impact on performance due to load redirection, we ran our algorithm on all 390 US locations to better understand the algorithm's behavior in a more comprehensive dataset. We assign each data center a pricing trace from Table I in a round-robin manner. While the algorithm doesn't account for redirection cost to a remote datacenter, we impose a redirection constraint using the radius parameter. Note that we use distance as a proxy for latency i.e. load see a higher latency if moved further away from the assigned



(a) January (b) July

Fig. 3: Average load movement with $r = 5000$ kms and TES capacity = 5 hours.



(a) r = 500 kms (b) r = 5000 kms

Fig. 4: Percentage of the load provided by OAC and TES with TES capacity = 5 hours.

location. We selected two months – January and July – as representative months to study the effect of load redirection due to OAC.

Figure 3 (a) and (b) shows the percentage of load redirected with TES capacity=5 hours and $r = 500$ and $r = 5000$ kms. We observe that 6% and 21% load is serviced locally with $r = 5000$ kms in January and July respectively. We note that although OAC may be abundant locally, our algorithm may redirect the load to another OAC feasible data center within the radius r parameter, or to a cheaper price location such that the energy cost is minimized. As our approach doesn't consider routing cost, with $r = 5000$ a higher redirection is expected. Indeed, redirecting load impacts performance and increases latency. However, delay-tolerant workloads such as batch jobs can be redirected to a remote data center to leverage OAC or cheaper electricity price.

To avoid redirection when local OAC is available, we ran a modified Lyapunov algorithm wherein we ensure all cooling energy OAC is provided locally and if OAC is infeasible locally, load is greedily redirected to a data center where OAC is feasible subject to the radius constraint. Finally, the residual load not satisfied by OAC is provided as input to our Lyapunov approach. We notice a 91% increase in load serviced locally in January, and a 36% increase in July. Thus, the modified algorithm minimally reroutes data as most of the cooling needs are provided locally. We omit results for $r = 500$ kms as all the load is serviced within 500 kms distance. Thus, performance is minimally impacted. While we use distance as a mechanism to restrict load movement, other attributes can be used to restrict load movement.

4) Does our approach use OAC and TES?: Finally, we analyze the extent to which our online approach uses OAC and TES when available. Due to existing chiller infrastructure provisioned in data centers, eliminating chillers altogether would be impractical. Nonetheless, future data centers may still benefit from using OAC and TES *both* environmentally and economically. Clearly, our goal doesn't maximize the use of TES or OAC but simply minimizes the cooling energy cost. However, reducing cost indirectly reduces chiller use. While OAC is "free", TES stores energy during cheaper price periods and discharges during higher price period. As a result, the algorithm leverages this property to utilize OAC and TES as much as possible.

We compute the percentage of cooling workload that is provided by OAC and TES for each month and a fixed TES capacity of 5 hours, a capacity large enough to shift load from on-peak price periods to off-peak price periods. Figure 4 (a) and (b), shows the percentage of the load provided by OAC and TES with $r = 500$ and $r = 5000$ kms. Note that our algorithm maximizes the use of OAC whenever available, fulfilling almost 99% of its cooling energy needs with OAC during winter. Even with a conservative $r = 500$, 57% and 15.8% cooling energy is provided by OAC and TES respectively. This indicates that CDNs can reduce its cooling energy footprint by switching to OAC. As can be seen in Figure 4, using OAC and TES in tandem is adequate to meet all the cooling energy needs. Additional cooling energy is provided by chillers.

VI. Related Work

Prior work on energy cost optimization focussed on leveraging price differential across various locations [13]. Other studies involved shutting down CDN servers to reduce server energy costs [14] [8]. Liu et. al. investigated the benefits of using renewable energy to reduce electricity cost and minimize use of brown energy [2]. In the context of thermal energy storage, researchers studied the use of thermal storage to reduce electricity cost in data centers [6]. Our Lyapunov-based technique is inspired by the approach used in Urgaonkar et. al. and Guo et. al. [3], [4] [12]. While Urgaonkar et. al. used Lyapunov optimization to reduce server energy cost for a single data center, Guo et. al. focussed on reducing energy cost for multiple data centers. Unlike their work, which is in the context of reducing server energy cost in data centers, our work focus on minimizing energy cost using OAC and TES. To the best of our knowledge, an integrated optimization of OAC with TES using a Lyapunov-based approach has not been studied previously.

VII. Conclusions

In this paper, we focussed on minimizing the cooling energy cost in CDNs by integrating OAC with TES. We provided a Lyapunov optimization-based online approach that optimally distributed load across geographically locations to maximize the use of OAC and optimally use TES. We empirically evaluated our approach using extensive traces and showed that our results are optimal. Our results showed that at least 64% and 98% cooling energy savings can be achieved in CDNs during summer and winter respectively. Also, we showed that CDNs can reduce its cooling energy footprint by at least 57% by switching to OAC.

References

[1] V. Mathew, R. K. Sitaraman, and P. J. Shenoy, "Reducing energy costs in internet-scale distributed systems using load shifting." in *COMSNETS*, 2014, pp. 1–8.

[2] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser, "Renewable and cooling aware workload management for sustainable data centers," in *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1. ACM, 2012, pp. 175–186.

[3] Y. Guo and Y. Fang, "Electricity cost saving strategy in data centers by using energy storage," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 6, pp. 1149–1160, 2013.

[4] R. Urgaonkar, B. Urgaonkar, M. J. Neely, and A. Sivasubramaniam, "Optimal power cost management using stored energy in data centers," in *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. ACM, 2011, pp. 221–232.

[5] *Designing a Very Efficient Data Center*, http://on.fb.me/1JRq3PK.

[6] Y. Wang, X. Wang, and Y. Zhang, "Leveraging thermal storage to cut the electricity bill for datacenter cooling," in *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*. ACM, 2011, p. 8.

[7] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE computer*, vol. 40, no. 12, pp. 33–37, 2007.

[8] M. Lin, A. Wierman, L. L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, no. 5, pp. 1378–1391, 2013.

[9] R. Miller, "Uptime institute: The average pue is 1.8," *Data Center Knowledge*, 2011.

[10] M. P. Tom Harvey and J. Bean, "Updated air-side free cooling maps: The impact of ASHRAE 2011 allowable ranges," *The Green Grid*, 2012, http://bit.ly/LfI14Q.

[11] S. Pelley, D. Meisner, T. F. Wenisch, and J. W. VanGilder, "Understanding and abstracting total data center power," in *Workshop on Energy-Efficient Design*, 2009.

[12] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan and Claypool Publishers, 2010.

[13] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 123–134.

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

[15] H. Xu, C. Feng, and B. Li, "Temperature aware workload management in geo-distributed datacenters," in *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*. ACM, 2013, pp. 373–374.

Appendix

Proof: (Theorem (4.1.1)) We first show (11) holds for $t = 0$. Using the definition of $X_i(t)$ from (7), we have:

$$Y_i(0) = Y_i^{max} = X(0) + \frac{VC_i^{max}}{\theta_i} + D_i^{max} \tag{15}$$

Thus, rearranging in terms of X_i, we have:

$$\begin{aligned} X(0) &= Y_i^{max} - \frac{VC_i^{max}}{\theta_i} - D_i^{max} \\ &\geq -\frac{VC_i^{max}}{\theta_i} - D_i^{max} \end{aligned} \tag{16}$$

Also,

$$\begin{aligned} X(0) &= Y_i^{max} - \frac{VC_i^{max}}{\theta_i} - D_i^{max} \\ &\leq Y_i^{max} - \frac{VC_i^{max}}{\theta_i} - D_i^{max} \end{aligned} \tag{17}$$

Thus, (7) holds for $t = 0$.

Now, suppose (7) holds for all slot t. We show that it holds for $t+1$. Suppose $X_i(t) \geq -\frac{VC_i^{min}}{\alpha_i\theta_i}$. We know that the coefficient of $R_i(t) \leq 0$ i.e. $-\alpha_i\theta_i X_i(t) - VC_i(t) \leq 0$. Thus, in order to maximize (10), $R_i(t)$ must be zero and $X_i(t)$ cannot increase. Next, if $X_i(t) \leq -\frac{VC_i^{min}}{\alpha_i\theta_i}$, then the maximum increase in $X_i(t)$ is R_i^{max}. Thus, in slot $t+1$ we have:

$$X_i(t+1) \leq -\frac{VC_i^{min}}{\alpha_i\theta_i} + R_i^{max} \tag{18}$$

Substituting the value of $\frac{VC_i^{min}}{\alpha_i}$ in (18) using (9), we get:

$$\begin{aligned} X_i(t+1) &\leq -\frac{\theta_i(Y_i - R_i^{max} - D_i^{max}) - VC_i^{max}}{\theta_i} + R_i^{max} \\ &\leq Y_i - \frac{VC_i^{max}}{\theta_i} - D_i^{max} \end{aligned} \tag{19}$$

This shows the bound on the right hand side of (11).

Now, we show (11) has a lower bound for all slots t. Suppose $X_i(t) \leq -\frac{VC_i^{max}}{\theta_i}$. We know that the coefficient of $D_i(t) \leq 0$ in (10) i.e. $\theta_i X_i(t) + VC_i(t) \leq 0$. Thus, in order to maximize (10), $D_i(t)$ must be zero and $X_i(t)$ cannot decrease. Next, if $X_i(t) \geq -\frac{VC_i^{max}}{\theta_i}$, then the maximum decrease in $X_i(t)$ is D_i^{max}. Thus, in slot $t+1$ we have:

$$X_i(t+1) \geq -\frac{VC_i^{max}}{\theta_i} - D_i^{max} \tag{20}$$

This shows the bound on the left hand side of (11). ■

Proof: (Theorem (4.1.2)) Suppose $Y_i(t) = Y^{max}$. Using (7),

$$X_i(t) = Y_i^{max} - \frac{V * C_i^{max}}{\theta_i} - D_i^{max} \tag{21}$$

We want to show that in (10), the coefficient multiplying $R_i(t) < 0$ i.e. $-\alpha_i\theta_i X_i(t) - VC_i(t) < 0$. Thus, in order to maximize (10), $R_i(t)$ must be zero. We have:

$$\begin{aligned} -\alpha_i\theta_i X_i(t) - VC_i(t) = -\alpha_i\theta_i Y_i^{max} + \alpha_i VC_i^{max} + \alpha_i\theta_i D_i^{max} \\ - VC_i(t) \end{aligned} \tag{22}$$

Now using the definition of θ_i, we get:

$$\theta_i(Y_i^{max} - D_i^{max}) = V(C_i^{max} - \frac{C_i^{min}}{\alpha_i}) + \theta_i R_i^{max} \tag{23}$$

Using this in (22), we get

$$\begin{aligned} -\alpha_i\theta_i X_i(t) - VC_i(t) &= -\alpha_i[V(C_i^{max} - \frac{C_i^{min}}{\alpha_i}) + \theta_i R_i^{max}] \\ &\quad + \alpha_i VC_i^{max} - VC_i(t) \\ &= VC_i^{min} - \alpha_i\theta_i R_i^{max} - VC_i(t) \\ &= -V(C_i(t) - C_i^{min}) - \alpha_i\theta_i R_i^{max} < 0 \end{aligned} \tag{24}$$

Thus, the optimal solution cannot charge when TES is full i.e. $Y_i(t) = Y^{max}$.

We now show that battery never discharges when TES is empty ie. $Y_i(t)=0$. Suppose $Y_i(t)=0$, we have:

$$X_i(t) = -\frac{VC_i^{max}}{\theta_i} - D_i^{max}. \tag{25}$$

We want to show that in (10), the coefficient multiplying $D_i(t) < 0$ i.e. $\theta_i X_i(t) + VC_i(t) < 0$. Thus, in order to maximize (10), $D_i(t)$ must be zero. . Now if we compute the term multiplying $D_i(t)$ in the objective, we get:

$$\begin{aligned}\theta_i X_i(t) + VC_i(t) &= -VC_i^{max} - \theta_i D_i^{max} + VC_i(t)\\ &= -V(C_i^{max} - C_i(t)) - \theta_i D_i^{max}\\ &< 0 \end{aligned} \tag{26}$$

Thus, the optimal solution cannot have any discharge when TES is empty. ■

Proof: (Theorem (4.1.3)) We use the Lyapunov optimization technique to show (13). We define the Lyapunov function: $L(X(t)) = \frac{1}{2}X^2(t)$. Also, the conditional 1-slot Lyapunov drift is defined as:

$$\Delta(X(t)) = \mathbb{E}\left\{L(X(t+1)) - L(X(t))|X(t)\right\} \tag{27}$$

Using (7), we know:

$$X_i(t+1) = X_i(t) + \alpha_i R_i(t) - D_i(t) \tag{28}$$

We now show that $\Delta X_i(t)$ is bounded. Squaring both sides, we get:

$$\begin{aligned}X_i^2(t+1) &= X_i^2(t) - 2X_i(t)[D_i(t) - \alpha_i R_i(t)]\\ &\quad + (D_i(t) - \alpha_i R_i(t))^2\\ \frac{X_i^2(t+1) - X_i^2(t)}{2} &= \frac{(D_i(t) - \alpha_i R_i(t))^2}{2}\\ &\quad - X_i(t)[D_i(t) - \alpha_i R_i(t)]\end{aligned}$$

Since $R_i(t) \le R_i^{max}$ and $D_i(t) \le D_i^{max}$, $\forall t$

$$\frac{(D_i(t) - \alpha_i R_i(t))^2}{2} \le \frac{(D_i^{max})^2 + (R_i^{max})^2}{2} = B_i \tag{29}$$

Taking conditional expectations on both sides given $X_i(t)$, we get:

$$\Delta X_i(t) \le B_i - X_i(t)\mathbb{E}\left\{D_i(t) - \alpha_i R_i(t)|X_i(t)\right\} \tag{30}$$

Following the Lyapunov optimization framework, we add to both sides of (30) the penalty term $\beta_i \mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\}$, where $\beta_i = \frac{Y_i^{max} - R_i^{max} - D_i^{max}}{C_i^{max} - \frac{C_i^{min}}{\alpha_i}}$

$$\begin{aligned}\Delta X_i(t) + \beta_i \mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\} &\le B_i + \beta_i \mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\}\\ &\quad - X_i(t)\mathbb{E}\left\{D_i(t) - \alpha_i R_i(t)|X_i(t)\right\}\end{aligned}$$

Using the equation $P_i(t) = W_i(t) + R_i(t) - D_i(t)$, we can rewrite the above equation as:

$$\begin{aligned}\Delta X_i(t) + \beta_i \mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\} &\le B_i + \beta_i \mathbb{E}\left\{W_i(t)C_i(t)|X_i(t)\right\}\\ &\quad + \beta_i \mathbb{E}\left\{R_i(t)C_i(t)|X_i(t)\right\} - \beta_i \mathbb{E}\left\{D_i(t)C_i(t)|X_i(t)\right\}\\ -X_i(t)\mathbb{E}\left\{D_i(t) - \alpha_i R_i(t)|X_i(t)\right\} &= B_i + \beta_i \mathbb{E}\left\{W_i(t)C_i(t)|X_i(t)\right\}\\ &\quad + \mathbb{E}\left\{R_i(t)[\alpha_i X_i(t) + \beta_i C_i(t)]|X_i(t)\right\}\\ &\quad - \mathbb{E}\left\{D_i(t)[X_i(t) + \beta_i C_i(t)]|X_i(t)\right\} = B_i\\ + \mathbb{E}\left\{\beta_i W_i(t)C_i(t)|X_i(t)\right\} &+ \mathbb{E}\left\{R_i(t)[\alpha_i X_i(t) + \beta_i C_i(t)]|X_i(t)\right\}\\ &\quad - \mathbb{E}\left\{D_i(t)[X_i(t) + \beta_i C_i(t)]|X_i(t)\right\}\end{aligned}$$

Using (9), we have $\beta_i = \frac{V}{\theta_i}$.

$$\begin{aligned}\theta_i \Delta X_i(t) + V\mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\} &\le B_i + \mathbb{E}\left\{VW_i(t)C_i(t)|X_i(t)\right\}\\ &\quad + \mathbb{E}\left\{R_i(t)[\alpha_i\theta_i X_i(t) + VC_i(t)]|X_i(t)\right\}\\ &\quad - \mathbb{E}\left\{D_i(t)[\theta_i X_i(t) + VC_i(t)]|X_i(t)\right\}\end{aligned}$$

Note that our online algorithm minimizes the last three terms for all data centers $i \in N$, over all possible slots t. This includes the offline solution to the problem. Indicating these terms by P^{lp}, we have:

$$\theta_i \Delta(X_i(t)) + V\mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\} \le B_i + V\mathbb{E}\left\{P_i^{lp}(t)C_i(t)|X_i(t)\right\}$$

Taking expectation on both sides, dividing both sides by V and summing over all data centers i and slots t, we have

$$\begin{aligned}\sum_{t=0}^{T}\sum_{i=1}^{N}\mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\} &\le \sum_{t=0}^{T}\frac{\sum_{i=1}^{N} B_i}{V}\\ + \sum_{t=0}^{T}\sum_{i=1}^{N}\mathbb{E}\left\{P_i^{lp}(t)C_i(t)|X_i(t)\right\} &+ \sum_{i=1}^{N}\frac{\mathbb{E}\left\{L(X_i(0)\right\}}{V}\\ &- \sum_{i=1}^{N}\frac{\mathbb{E}\left\{L(X_i(T)\right\}}{V}\end{aligned}$$

Dividing both sides with T and taking limit as $T \to \infty$:

$$\begin{aligned}\lim_{t\to\infty}\frac{1}{T}\sum_{t=0}^{T}\sum_{i=1}^{N}\mathbb{E}\left\{P_i(t)C_i(t)|X_i(t)\right\} &\le \frac{\sum_{i=1}^{N} B_i}{V}\\ &+ \lim_{t\to\infty}\frac{1}{T}\sum_{t=0}^{T}\sum_{i=1}^{N}\mathbb{E}\left\{P_i^{lp}(t)C_i(t)|X_i(t)\right\}\end{aligned}$$

where $\mathbb{E}\left\{L(X_i(0)\right\}$ and $\mathbb{E}\left\{L(X_i(T)\right\}$ is finite and non-negative. This shows (13). ■