



Exploiting Temporal Diversity of Water Efficiency to Make Data Center Less “Thirsty”

Mohammad A. Islam, Kishwar Ahmed, Shaolei Ren, and Gang Quan,
Florida International University

<https://www.usenix.org/conference/icac14/technical-sessions/presentation/islam>

**This paper is included in the Proceedings of the
11th International Conference on Autonomic Computing (ICAC '14).
June 18–20, 2014 • Philadelphia, PA**

ISBN 978-1-931971-11-9

**Open access to the Proceedings of the
11th International Conference on
Autonomic Computing (ICAC '14)
is sponsored by USENIX.**

Exploiting Temporal Diversity of Water Efficiency to Make Data Center Less “Thirsty”

Mohammad A. Islam, Kishwar Ahmed, Shaolei Ren, Gang Quan
Florida International University

Abstract

Data centers, which include both cyber (e.g., servers) and physical (e.g., cooling units) assets, are notorious for their energy consumption and carbon footprint. Nonetheless, a less-known fact about data centers is that they are extremely “thirsty” (for cooling), consuming millions of gallons water each day and raising serious concerns amid extended droughts. To curtail the surging water footprint, we adopt a holistic cyber-physical approach and incorporate the inherent physical characteristic of data center — time-varying water efficiency — into server provisioning and workload management. Specifically, we propose an online batch job scheduling algorithm, called WACE (minimization of WATER, Carbon and Electricity cost), which dynamically adjusts server provisioning to reduce the water consumption by deferring delay-tolerant batch jobs to water-efficient time periods. We demonstrate the effectiveness of WACE via trace-based simulations, showing that WACE reduces 27% water consumption compared to state-of-the-art scheduling algorithms.

1 Introduction

Ubiquitous Internet services and explosive IT demand have led to a new wave of constructing gigantic data centers, accounting for 1.7-2.2% of the total electricity usage in the United States as of 2010 [13]. Data centers consist of both cyber assets (e.g., servers, networking equipment) and physical assets (e.g., cooling systems, energy storage device). While data centers are notorious for huge energy consumption due to power-hungry servers, cooling systems — data centers’ physical assets — are very “thirsty”, evaporating millions of gallons of water each day for rejecting server heat. For example, cooling towers in AT&T’s large data center facilities consume 1 billion gallons of water in 2012, approximately 30% of the entire company’s water consumption [3]. In addition, just as they are accountable for carbon emissions via electricity usage, data centers also consume a vast amount of offsite water *remotely* embedded in electric-

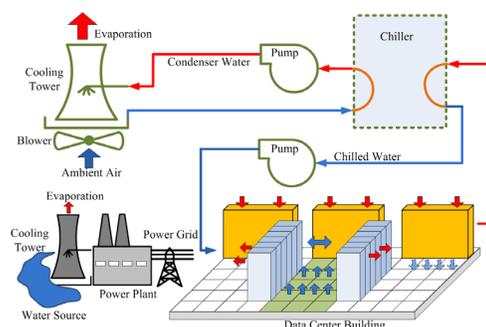


Figure 1: Water consumption in data center.

ity generation: e.g., in the U.S., an average of 1.8 liter of water is evaporated, or “lost”, into the air for just one kilowatt-hour electricity generation, even excluding the water-consuming hydropower [29, 32]. A typical water-cooled data center is illustrated in Fig. 1.

Why is water critical to data centers? As shown in a recent survey by Uptime Institute [34], cooling towers are widely employed by large data centers (over 40%), even though there exist various types of cooling systems and data centers in low-temperature regions may use cold outside air for cooling. Water conservation has become essential for green certifications (being sought by a majority of large data centers [34]), tax credits [36], and corporate social responsibility [3]. On the other hand, water is also crucial for electricity generation (e.g., thermoelectricity, nuclear power) [29, 32], which is undeniably essential for data center operation.

Water is not equal to energy. Despite the nexus between water and energy [29, 32], the existing studies for minimizing data center energy consumption cannot minimize water footprint, because when optimizing for energy efficiency, they neglect the physical characteristic (in particular, *time-varying* water efficiency, details provided in Section 3) of data center cooling systems and electricity generation. In fact, for reducing water footprint, it is

not only important to minimize energy, but also crucial to consider “when” to consume energy.

Prior studies on minimizing electricity cost [26] and carbon emissions [9] do *not* lead to water minimization solutions either, because water efficiency is not in proportion to electricity cost-/carbon-efficiency (e.g., nuclear power incurs little carbon emission but can consume more than 2L of water per kWh [20, 23]). While using air economizer (i.e., “free” cooling) and recycled/sea water can reduce potable water consumption [10, 21], these techniques, however, focus on improved “engineering” and do not apply to all data centers, because they typically require high upfront costs and/or suitable location-/climate conditions.

Software-based approach to water conservation.

We incorporate water footprint as an integral metric into data center operation. For water conservation, the high-level intuition is to exploit the inherent physical characteristic of data center onsite cooling towers and offsite electricity generation (i.e., temporal diversity of water efficiency): we would like to defer batch workloads to time periods with better water efficiency while shutting down some servers during time periods with low water efficiency. To turn this intuition into reality, a notable challenge is that it is difficult to determine which time periods are water-efficient without foreseeing far future information, due to the time-varying nature of water efficiency, job arrivals, carbon emission rate and electricity price. A straw man technique can be setting a threshold on water efficiency: process batch jobs only when data center water efficiency is better than the threshold. Nonetheless, setting a too high threshold may degrade the delay performance too much, whereas setting a too low threshold may unnecessarily waste water.

To address the challenge, we propose a new online delay-tolerant batch workload management algorithm, called WACE (minimization of WAter, Carbon and Electricity cost), to reduce water footprint, while also including electricity cost and carbon footprint as an integral part of the optimization objective. A remarkable feature of WACE is that it can be implemented online based on the currently available information, yet we demonstrate its effectiveness through a traced-based simulation. Our simulation results show that, compared with state-of-the-art scheduling algorithms, WACE can reduce the cost by approximately 20%, while reducing the water consumption by approximately 27%.

2 Model

We consider a discrete-time model by equally dividing the entire time horizon of interest (e.g., one year) into K time slots. The duration of each time slot may range from minutes up to an hour. We focus on facility-level server provisioning and workload management. Next,

we provide modeling details, which are consistent with the literature (e.g., [16, 38]).

2.1 Workload

In general, there are two types of workloads in data centers: delay-tolerant batch workloads (e.g., back-end processing, scientific applications) and delay-sensitive interactive workloads (e.g., web services or business transactional applications). We focus on scheduling batch workloads and denote by $a(t) = [0, a_{max}]$ the amount of batch workload arrivals at time t , quantified in terms of machine-time [17, 35]. Although this widely-used model cannot capture all the low-level details (e.g., parallelism), it provides a *good* guidance for dynamically “sizing” the data center (i.e., how many servers can be turned off) and hence suffices for our purpose.

2.2 Data center

The data center has $r(t)$ amount of on-site renewable energy, e.g., by solar panels [1]. There are a total of $M(t)$ homogeneous servers that are available for processing batch jobs at time t . Servers may run at different processing speeds and incur different power [18]: we consider an array of finite processing speeds denoted by $\mathcal{S} = \{s_1, \dots, s_N\}$, from which a speed s is chosen for processing batch workloads. Following [8, 18], we express the average power consumption of a server at time t as $\alpha \cdot s(t)^n + p_0$, where α is a positive factor and relates the processing speed to the power consumption, n is empirically determined (e.g., between 1 \sim 3), and p_0 represents the power consumption in idle or static state. We model the server energy consumption by interactive workloads as an exogenously-determined value $p_{int}(t)$. We write the energy consumption by batch workloads as $p_{bat}(t) = m(t) \cdot [\alpha \cdot s(t)^n + p_0]$. Hence, the total server energy consumption can be formulated as

$$p(t) = p_{bat}(t) + p_{int}(t). \quad (1)$$

Next, given the available on-site renewable energy $r(t)$, the data center’s electricity usage at time t is $[\gamma(t)p(t) - r(t)]^+$, where $[\cdot]^+ = \max\{\cdot, 0\}$ and $\gamma(t)$ is the factor of Power Usage Effectiveness (PUE) capturing the non-IT energy consumption.

3 Online Batch Job Scheduling: WACE

In this section, we formulate the cost, present problem formulation and develop an online algorithm, WACE, to minimize the total cost via online batch job scheduling.

3.1 Cost

Our work aims to address three “costs”: water consumption, electricity cost and carbon emission.

- **Water consumption.** As illustrated in Fig. 1, water is consumed in data center’s onsite physical asset (i.e.,

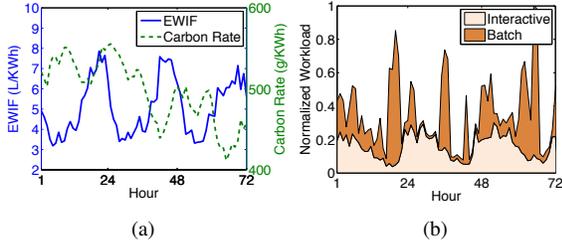


Figure 2: Water-carbon efficiency and workload trace. **(a)** EWIF and carbon emission rate in California. **(b)** Workload trace [12, 16].

cooling tower) and also offsite power plants. To assess water usage efficiency, an emerging metric, called Water Usage Effectiveness (WUE), was recently developed by The Green Grid [32]. WUE is the ratio of water consumption to IT equipment energy, where water consumption includes both direct and indirect water consumption (i.e., onsite water for data center cooling and offsite water for electricity production).

Direct water: Cooling towers directly consume onsite fresh water, and direct WUE at time t , denoted by $\varepsilon_D(t)$, is affected by various factors, such as non-stationary outside wet bulb temperature [31]. Hence, as an inherent characteristic of cooling tower, direct WUE exhibits a time-varying nature. In practice, direct WUE can be monitored in real-time [7].

Indirect water: Indirect water efficiency is quantified in terms of Energy Water Intensity Factor (EWIF), which measures the amount of water consumption per kWh electricity. Different energy fuel types (e.g., thermal, nuclear, hydro) have different EWIFs [19]. As the energy fuel mixes vary over time due to varying peak/non-peak demand [9], the resulting average EWIF exhibits a temporal diversity. Fig. 2(a) demonstrates the time-varying EWIF for California, calculated based on [19] and California energy fuel mixes [2]. In our study, we calculate the average EWIF as follows:

$$\varepsilon_I(t) = \frac{\sum_k b_k(t) \times \varepsilon_k}{\sum_k b_k(t)} \quad (2)$$

where $b_k(t)$ denotes the amount of electricity generated from fuel type k , and ε_k is the EWIF for fuel type k .

Now, we formulate the water consumption at time t as

$$w(t) = \varepsilon_D(t) \cdot p(t) + \varepsilon_I(t) \cdot [\gamma(t) \cdot p(t) - r(t)]^+, \quad (3)$$

where $p(t)$ is the server power, $\gamma(t)$ is PUE and $r(t)$ is available on-site renewable energy.

• **Electricity cost.** We denote the electricity price at time t by $u(t)$, and hence the electricity cost is $e(t) = u(t) \cdot [\gamma(t)p(t) - r(t)]^+$, where $[\gamma(t)p(t) - r(t)]^+$ is the

data center electricity usage. Note that with the emergence of smart grid, large data centers may have the market power to influence real-time electricity price, and if so, the electricity price $u(t)$ can be modeled following [37, 39].

• **Carbon emission.** We calculate the (average) carbon emission rate following [9], and Fig. 2(a) demonstrates the time-varying carbon emission rate for California. An interesting observation is that carbon emission efficiency does not align with EWIF (i.e., indirect water efficiency). The difference between carbon efficiency and water efficiency becomes even greater if we factor in the time-varying direct WUE for cooling towers. The same observation also holds for electricity cost efficiency versus water efficiency. Next, we express the total carbon footprint of the data center at time t as $c(t) = \phi(t) \cdot [\gamma \cdot p(t) - r(t)]^+$, where we neglect the small carbon emission by the on-site renewable energy.

3.2 Problem formulation

In this subsection, we first describe our objective and constraints, and then present the problem formulation.

Objective. We aim to minimize the electricity cost while incorporating carbon emission and water consumption. We construct a parameterized *total cost* function as follows

$$g(t) = e(t) + h_w \cdot w(t) + h_c \cdot c(t), \quad (4)$$

where $h_w \geq 0$ and $h_c \geq 0$ are weighting parameters for water consumption and carbon emission relative to the electricity cost. Such a multi-objective formulation is common in the literature. Our optimization objective is to minimize the long-term average cost expressed as $\bar{g} = \frac{1}{K} \sum_{t=0}^{K-1} g(t)$, where K is the total number of time slots in the period of interest.

Constraints. First, the number of servers to process batch jobs needs to satisfy

$$0 \leq m(t) \leq M(t), \quad (5)$$

where $M(t)$ is the maximum available number of servers. The server can only select one of the supported speeds:

$$s(t) \in \mathcal{S} = \{s_0, s_1, \dots, s_N\}. \quad (6)$$

We also need to guarantee that batch jobs will be processed (without dropping):

$$\bar{a} < \bar{b}, \quad (7)$$

$$b(t) = m(t) \cdot s(t), \quad (8)$$

where $\bar{a} = \sum_{t=0}^{K-1} a(t)$ and $\bar{b} = \sum_{t=1}^{K-1} b(t)$ are the long-term average workload arrival and allocated server capacity, respectively. The constraint (8) states the relation between processed batch jobs and server provisioning.

Algorithm 1 WACE

- 1: At the beginning of each time t , observe the data center state information $r(t)$, $\varepsilon_D(t)$, $\varepsilon_I(t)$, $\phi(t)$ and $p_B(t)$, for $t = 0, 1, 2, \dots, K - 1$
- 2: Choose $s(t)$ and $m(t)$ subject to (5)(6)(8) to minimize

$$V \cdot g(t) - q(t) \cdot b(t) \quad (12)$$

- 3: Update $q(t)$ according to (11).
-

Problem formulation. We present an offline problem formulation for batch job scheduling as follows

$$\mathbf{P1}: \quad \min_{\mathcal{D}} \bar{g} = \frac{1}{K} \sum_{t=0}^{K-1} g(s(t), m(t)) \quad (9)$$

$$s.t., \quad \text{constraints (5), (6), (7), (8)}. \quad (10)$$

Clearly, finding the optimal offline solution to **P1** requires complete offline information (i.e., workload arrivals, direct WUE, EWIF, carbon emission rate, on-site renewables and electricity prices) throughout the entire time period, which is very challenging, if not impossible, to obtain in practice. Therefore, we develop an online algorithm below.

3.3 WACE

To enable an online algorithm, we remove (7) and maintain a batch job queue that stores unfinished batch jobs. Specifically, assuming that $q(0) = 0$, we write the job queue dynamics as

$$q(t+1) = [q(t) - b(t)]^+ + a(t), \quad (11)$$

where $[\cdot]^+ = \max\{\cdot, 0\}$, $a(t)$ quantifies batch job arrivals, and $b(t)$ indicates the amount of processed jobs.

Intuitively, when the queue length becomes large, the data center should increase the number of servers and/or server speed to reduce the queue backlog to avoid too much delay. Hence, we incorporate the queue length into the objective function, as described in Algorithm 1. In (12), the queue length determines how much emphasis the optimization gives on the resource provisioning $b(t)$ for processing batch jobs. WACE is purely online and only requires the currently available information. The parameter $V \geq 0$ in line 2 of Algorithm 1, referred to cost-delay parameter, acts as a tradeoff control knob: the larger V , the smaller impact of the queue length on optimization decisions.

While Algorithm 1 appears simple, it is provably efficient, even compared to the optimal offline algorithm that has future information. In particular, one can show based on the recently-developed Lyapunov technique [22] that the gap between the average cost achieved

by WACE and that by the optimal offline algorithm is bounded, while the batch job queue length is also upper bounded, translating into a finite queueing delay. We omit the proof details due to space limitations.

4 Performance Evaluation

This section presents trace-based simulation studies of a data center to evaluate WACE.

4.1 Data Sets

We consider a large data center consisting of 300,000 homogeneous servers with a peak power of 64MW. Each server has 15 discrete speed levels, uniformly ranging from 1.6GHz to 3GHz. The duration of each time slot is set to 1 hour and the total simulation period is 1 year. The default weighting parameters for water consumption and carbon emission are $h_w = 15$ and $h_c = 0.15$, respectively, and the PUE is set to 1.2.

- **Workloads:** We consider that the data center serves both interactive and batch jobs, which are taken from the literature [12, 16], respectively. The maximum arrival rate for the batch jobs and interactive jobs are scaled to be 80% and 30% of the data center maximum capacity, respectively, while the maximum combined workload arrival rate still satisfies the peak data center capacity. Fig. 2(b) illustrates a snapshot of the traces for 3 days, normalized with respect to the maximum data center capacity.

- **Others:** We use the demand-responsive electricity prices modeled by the fitted function shown in [37]. We collect the temperature data in Mountain View, CA, and fuel mix data of California ISO [2, 5] from October, 2012 to September, 2013. We use the EWIF and carbon emission rates for different electricity generation methods presented in [19, 30, 33] to calculate the EWIF and carbon emission rate for the data center. The first 3-day data for EWIF and carbon rate are shown in Fig. 2(a). Direct WUE is modeled based on empirical measurement [31].

4.2 Simulation Results

We now compare the performance of WACE with three benchmarks.

Benchmarks

The three benchmarks are described as follows.

- **SAVING:** SAVING only optimizes the electricity cost of the data center and is water- and carbon-oblivious. It applies WACE with zero weights for water and carbon.

- **CARBON:** CARBON only optimizes the carbon emission of the data center and is electricity- and water-oblivious. Essentially, it applies WACE with an “infinite” weight for carbon.

- **ALWAYS:** ALWAYS does not use any optimization and tries to process jobs as soon as possible. This is the *de-factor* algorithm used in many data centers.

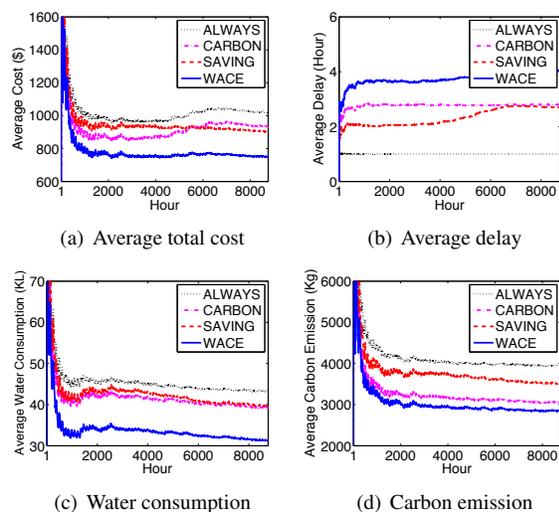


Figure 3: Comparison between WACE and benchmarks.

Performance comparison

Here, by fixing the cost-delay parameter V , we compare the performance of WACE with the three benchmark algorithms and show the results in Fig. 3, where the average value at any time slot t represents the cumulative average from 0 to t . In Fig. 3(a), we see that WACE achieves the lowest average total cost among all the algorithms. Compared to WACE, the other three algorithms, i.e., SAVING, CARBON and ALWAYS, incur a 20%, 24% and 36% higher cost, respectively. The delay performance comparison shows that ALWAYS has the lowest delay of 1 (due to its greedy nature), both SAVING and CARBON have an average delay around 3 time slots (i.e., hours), while WACE has a delay close to 4 time slots. The delay figure identifies that WACE is taking more advantage from the delay tolerance of batch jobs and hence achieves a lower average total cost by opportunistically processing batch jobs when the combined cost factor is relatively lower. The water consumption and carbon emission results show that compared to WACE, the benchmark algorithms, i.e., SAVING, CARBON and ALWAYS, incur more water consumption by 27%, 25% and 38.5% and higher carbon emission by 23%, 7.4% and 39%, respectively. This highlights the benefit of WACE in terms of sustainability (while the delay performance is compromised to a small and tolerable extent).

Impact of water and carbon weights

Now, we will study the impact of water weight (h_w) and carbon weight (h_c) on the performance of WACE. For both weighting factors, we start from zero and go up to the value which makes the water/carbon cost equal to 90% of the total cost, while keeping the other weight at

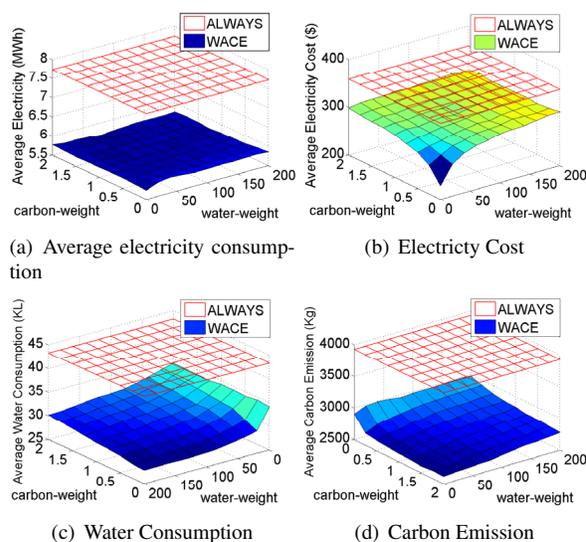


Figure 4: Impact of water and carbon weights.

its default value (i.e., $h_w = 15$, $h_c = 0.15$). We show a set of 3-dimension figures to capture all the possible combinations of water and carbon weights within the above mentioned range, and compare WACE with ALWAYS (which is the *de facto* reference algorithm, as currently many data centers are still performance-driven). In all cases, V is appropriately chosen for WACE to achieve an average delay equal to 4 hours.

Fig. 4(a) shows that the average electricity consumption remains almost same with varying water and carbon weights. This is because, the actual energy consumption for processing a fixed amount of workloads remains relatively the same, no matter WACE is trying to reduce carbon emission (i.e., high value of carbon weight) or water consumption (i.e., high value of water weight). The small variation in electricity consumption, however, can be attributed to the effect of discrete speed settings, which let servers run with variable dynamic energy consumption (but still fixed static energy as long as a server is turned on). From Fig. 4(b), we see that increase in either water or carbon weight increases the electricity cost. We have already seen that the weighting factors have little effect on the actual energy consumption, and hence the increased electricity cost implies the following fact: with increased water and/or carbon weight, WACE schedules batch jobs to find low water consumption and/or carbon emission due to sustainability considerations, not solely caring about the electricity cost. Fig. 4(c) and Fig. 4(d) show the decreasing trend of water consumption and carbon emission as the corresponding weighting factor is increased. The effect is straightforward, as increased weighting factor means a higher priority in the optimization algorithm.

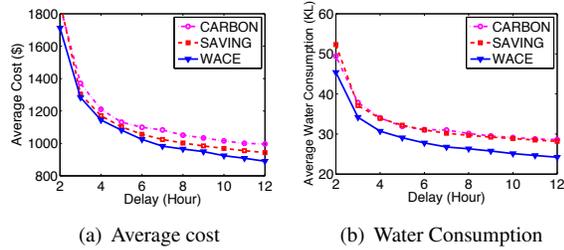


Figure 5: Impact of average delay constraint.

Next, we note that WACE has a lower electricity cost, carbon emission and water consumption compared to ALWAYS for a wide range of water and carbon weights, although no algorithms (including WACE) can possibly outperform SAVING/CARBON in terms of electricity cost/carbon emissions, as they solely minimize their respective metric. Nonetheless, Fig. 4 shows that, by appropriately choosing water and carbon weights, we will get a better performance from WACE than the widely-employed ALWAYS scheduling algorithm in terms of electricity cost, water consumption and carbon emissions (at the expense of increasing delay for batch jobs).

Impact of average delay constraint

Now, we study the relation of total cost and water with average delay performance. For this purpose, we vary the delay constraint from 2 to 12 time slots and show the corresponding cost and water consumption in Fig. 5. We see in Fig. 5(a) that the average total cost decreases for all algorithms with increased delay constraint (i.e., a less stringent delay requirement). In particular, given any delay constraint, WACE has the lowest average total cost. The performance gap between WACE and other two algorithms (i.e., SAVING and CARBON) increases with more relaxed delay constraint. ALWAYS is not shown as its delay is constantly 1 time slot. Fig. 5(b) shows a similar pattern of the change in water consumption with varying delay constraints. We can see that WACE has the lowest water consumption as it incorporates time-varying water efficiency when making scheduling decisions. Fig. 5 provides us with an important guidance for choosing an appropriate set of water and carbon weights so that the data center can run in low cost and/or reduced water footprints without much impact on the average delay performance.

We also conduct sensitive studies to demonstrate the robustness of WACE, and interested readers are referred to the technical report [12].

5 Related Work

In this section, we discuss the related work.

- **Data center optimization:** Several prior studies have focused on identifying methods of cost cutting

while ensuring the quality-of-service. For example, finding a balance between energy cost of data center and performance loss through dynamically provisioning server capacity has been the primary focus of many recent studies [11, 16]. Other approaches include, but are not limited to, exploiting the spatio-temporal variation of electricity prices [15, 25, 26]. Cyber-physical approaches to optimizing data center cooling system and server management are also investigated [14, 24]. Electricity cost can be further reduced when the advantage of geographical load balancing is combined with the dynamic capacity provisioning approach [?, 26]. Nonetheless, none of these studies address water consumption in data centers.

- **Water reduction in data center:** Most of the existing efforts on water efficiency have been focusing on improved “engineering”: for example, installing advanced cooling system [4], and using recycled water [10]. Our study focuses on integrating physical characteristic of time-varying water efficiency with control of data center’s cyber asset (i.e., servers and workloads). More recent study [28] aims at optimizing water efficiency for delay-sensitive interactive workloads, but it does not apply to our study, because it neglects carbon footprints and does not exploit the temporal diversity of water efficiency. Another work [27] preliminarily minimizes water footprint via resource management, but it neglects many important factors, such as carbon emissions, demand-responsive electricity prices, discrete server speed selection and interactive jobs. Other research that is remotely related to data center water consumption includes [6], which develops a dashboard to visualize the water efficiency. To our best knowledge, holistically minimizing electricity cost, carbon emission and water footprint by leveraging the delay tolerance of batch jobs and temporal diversity of water efficiency has not been studied by any prior work.

6 Conclusions

In this paper, we studied water consumption in data centers and proposed an efficient online batch job scheduling algorithm, WACE, for minimizing the operational cost (incorporating electricity cost, water consumption and carbon emission) while bounding the average queue length. WACE exploits and integrates the physical characteristic of time-varying water efficiency with data center’s cyber asset management (i.e., server provisioning and workload scheduling). We demonstrated the effectiveness of WACE via trace-based simulations, showing that WACE reduces the water consumption by over 27% compared to the state-of-the-art solutions, with a negligible delay increase.

References

- [1] Apple and the environment, <http://www.apple.com/environment/>.
- [2] California ISO, <http://www.caiso.com/>.
- [3] AT&T Sustainability, <http://www.att.com/gen/landing-pages?pid=24188>.
- [4] Facebook data center dashboard, <http://www.fbpuewue.com>.
- [5] Historical Weather, <http://www.wunderground.com/>.
- [6] C. Bash, T. Cader, Y. Chen, D. Gmach, R. Kaufman, D. Milojicic, A. Shah, and P. Sharma. Cloud sustainability dashboard, dynamically assessing sustainability of data centers and clouds. *HP Labs Tech. Report (HPL-2011-148)*, Sep. 2011.
- [7] Facebook. Open sourcing pue, wue dashboards, <https://code.facebook.com/posts/272417392924843/open-sourcing-pue-wue-dashboards>.
- [8] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *ISCA*.
- [9] P. X. Gao, A. R. Curtis, B. Wong, and S. Keshav. It's not easy being green. *SIGCOMM Comput. Commun. Rev.*, 42(4):211–222, Aug. 2012.
- [10] Google's Data Center Efficiency. <http://www.google.com/about/datacenters/>.
- [11] B. Guenter, N. Jain, and C. Williams. Managing cost, performance and reliability tradeoffs for energy-aware server provisioning. In *IEEE Infocom*, 2011.
- [12] M. A. Islam, K. Ahmed, S. Ren, and G. Quan. Making data center less “thirsty” via online batch job scheduling. *Tech. Report*, 2014, http://users.cis.fiu.edu/~sren/doc/paper/icac_2014_water_full.pdf.
- [13] J. G. Koomey. Growth in data center electricity use 2005 to 2010, 2011.
- [14] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos. Thermocast: A cyber-physical forecasting model for datacenters. In *KDD*, 2011.
- [15] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew. Online algorithms for geographical load balancing. In *IGCC*, 2012.
- [16] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *IEEE Infocom*, 2011.
- [17] 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 *SIGMETRICS*, 2012.
- [18] J. R. Lorch and A. J. Smit. Improving dynamic voltage scaling algorithms with pace. In *SIGMETRICS*, 2001.
- [19] J. Macknick, R. Newmark, G. Heath, and K. Hallett. A review of operational water consumption and withdrawal factors for electricity generating technologies. *NREL Tech. Report: NREL/TP-6A20-50900*, 2011.
- [20] U.S. Dept. of Energy. Energy demands on water resources. Dec. 2006.
- [21] Microsoft Global Foundation Services. Microsoft recycles waste to provide clean power for data center R&D, 2012 (<http://www.globalfoundationservices.com>).
- [22] M. J. Neely. *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool, 2010.
- [23] Nuclear Energy Institute. Environment: Emissions prevented, http://www.nei.org/resourcesandstats/nuclear_statistics/Environment-Emissions-Prevented.
- [24] L. Parolini, N. Tolia, B. Sinopoli, and B. H. Krogh. A cyber-physical systems approach to energy management in data centers. In *ICCPs*, 2010.
- [25] A. Qureshi, R. Weber, H. Balakrishnan, J. Gutttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *SIGCOMM*, 2009.
- [26] L. Rao, X. Liu, L. Xie, and W. Liu. Reducing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *IEEE Infocom*, 2010.
- [27] S. Ren. Batch job scheduling for reducing water footprints in data center. In *Allerton*, 2013.
- [28] S. Ren. Optimizing water efficiency in distributed data centers. In *Cloud and Green Computing*, 2013.
- [29] M. J. Rutberg. Modeling water use at thermoelectric power plants, 2012.

- [30] J. V. Spadaro, L. Langlois, and B. Hamilton. Greenhouse gas emissions of electricity generation chains: Assessing the difference. *IAEA bulletin*, 42(2):19–28, 2000.
- [31] SPX Cooling. Water usage calculator, <http://spxcooling.com/green/leed/water-usage-calculator/>.
- [32] The Green Grid. Water usage effectiveness (WUE): A green grid data center sustainability metric. *Whitepaper*, 2011.
- [33] P. A. Torcellini, N. Long, and R. Judkoff. Consumptive water use for us power production, 2003.
- [34] Uptime Institute. Data center industry survey, 2013, <http://uptimeinstitute.com/2013-survey-results>.
- [35] R. Uргаonkar, B. Uргаonkar, M. J. Neely, and A. Sivasubramaniam. Optimal power cost management using stored energy in data centers. In *SIGMETRICS*, 2011.
- [36] U.S. Green Building Council. Leadership in energy & environmental design, <http://www.usgbc.org/leed>.
- [37] P. Wang, L. Rao, X. Liu, and Y. Qi. D-pro: Dynamic data center operations with demand-responsive electricity prices in smart grid. *IEEE Transactions on Smart Grid*, 3(4):1743–1754, December 2012.
- [38] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. J. Neely. Data centers power reduction: A two time scale approach for delay tolerant workloads. In *Infocom*, 2012.
- [39] Y. Zhang, Y. Wang, and X. Wang. Electricity bill capping for cloud-scale data centers that impact the power markets. In *ICPP*, 2012.