This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

Water-Constrained Geographic Load Balancing in Data Centers

Mohammad A. Islam, Shaolei Ren, Gang Quan, Muhammad Z. Shakir, Athanasios V. Vasilakos

Abstract—Spreading across many parts of the world and presently hard striking California, extended droughts could even potentially threaten reliable electricity production and local water supplies, both of which are critical for data center operation. While numerous efforts have been dedicated to reducing data centers' energy consumption, the enormity of data centers' water footprints is largely neglected and, if still left unchecked, may handicap service availability during droughts. In this paper, we propose a water-aware workload management algorithm, called WATCH (WATer-constrained workload sCHeduling in data centers), which caps data centers' long-term water consumption by exploiting spatio-temporal diversities of water efficiency and dynamically dispatching workloads among distributed data centers. We demonstrate the effectiveness of WATCH both analytically and empirically using simulations: based on only online information, WATCH can result in a provably-low operational cost while successfully capping water consumption under a desired level. Our results also show that WATCH can cut water consumption by 20% while only incurring a negligible cost increase even compared to state-of-the-art cost-minimizing but water-oblivious solution. Sensitivity studies are conducted to validate WATCH under various settings.

Index Terms—Data center, Geographical load balancing, Resource management, Sustainable IT, Water footprint.

1 Introduction

Extended droughts are becoming a norm worldwide. For example, California has been experiencing its fourth year of drought in a row, mandating water restrictions throughout the state [3]. Meanwhile, drought is emerging as a hidden threat to many industry sectors, including data centers.

Electricity production. Data centers have a gigantic appetite for electricity. Nonetheless, extended droughts and water shortage are threatening reliable electricity production (e.g., in Texas and California [36] which are also major markets of data centers), because electricity production, especially thermoelectric and nuclear power, consumes an astonishing amount of water in the power plant through steam condensation (i.e., water evaporates from cooling towers into the environment) [32]. For example, the U.S. national average water consumption reaches 1.8 liters water per kilowatt-hour electricity (L/kWh), excluding the even more water-consuming hydropower [45].

Cooling system. While advanced cooling systems (e.g., air economizer [10]) are good at water saving, most data centers are located in places where installation of such cooling systems is not feasible or economical. Thus, it is common that data centers, such as AT&T and eBay [5], [7], rely on water-intensive methods for cooling (e.g., water-side economizer and water-cooled chillers) [46]. It was estimated that a 15MW data center could consume 360,000 gallons of cooling water each day [8], while another report [26] said that the U.S. National Security Data center in Utah would require up to 1.7 million gallons of water for cooling each day (enough to satiate 10,000 households' daily needs).

Even in regions with relatively abundant water, there are strong motivations for data centers to conserve water. For example, reducing water by 10-25% is a prerequisite for green certifications (e.g. LEED program [48]) which provide tax/zoning benefits and are being actively pursued by 77% of large data centers as shown in a recent survey [46]. Water compliance codes are tightening in many regions [3], and the U.S. government requires all federal facilities to reduce water usage by 2% each year through 2020 [6]. Last but not least, forward-looking companies have been taking active steps to conserve water for mitigating business risks, improving public image and fulfilling social responsibility (e.g., AT&T's recent efforts [7]).

Limitations of the existing research. Although water footprint is surfacing as a critical concern, it has been rarely studied for data centers. The recent progress towards energy/cost/carbon reduction [20], [27], [29], [30], [40] turns out to be inadequate for water conservation. As specified in Section 2, this is because data center water usage effectiveness (i.e., water consumption per unit IT energy [45]) changes over location and also over time in its own way: the same amount of energy but consumed at different times/locations may result in different water footprints.

The latest efforts to reduce water consumption in data centers have primarily focused on facility or infrastructure upgrades (i.e., improved "engineering"), such as using recycled water or seawater instead of potable water, using chemicals to reduce water "blown down" in cooling towers, and directly using outside cold air [7], [10], [16], [22]. These techniques, however, often require high upfront costs and/or suitable locations/climate conditions (which may not be satisfied by drought areas). Moreover, the offsite

- *Mohammad A. Islam, Shaolei Ren, and Gang Quan are with Florida International University, Miami, FL, 33199. E-mail: {misla012, sren, gang.quan}@fiu.edu*
- *Muhammad Z. Shakir is with Texas A&M University at Qatar. E-mail: muhammad.shakir@qatar.tamu.edu*
- *Athanasios V. Vasilakos is with Lulea University of Technology, Sweden. E-mail: athanasios.vasilakos@ltu.se*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

water usage at power plants, which is attributed to data centers [20], [45], is still left unaddressed.

Proposed approach. Recognizing the critical importance of conserving water for data centers, we focus on a set of geo-distributed data centers and study: how to minimize data centers' operational cost while surviving drought by keeping the long-term overall water footprint under a cap?[1] We choose to cap the *long*-term water footprint rather than purely minimizing it, because survival of drought clearly requires persistent efforts and water capping matches the current practice to achieve water conservation (e.g., industry goal [7], water rationing, government regulations [6], green certifications [48]). Instead of tackling the problem using improved "engineering", we propose a software-based approach by optimizing workload management. Our approach relies on the following two techniques.

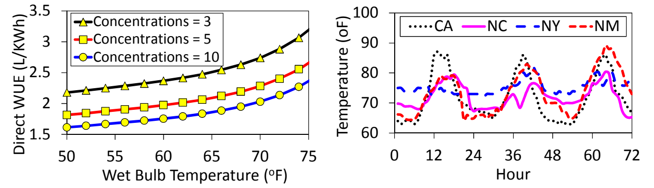
- Geographic load balancing (GLB): dynamically dispatching workloads to geo-distributed data centers.
- Power proportionality: dynamically turning on/off servers in accordance with workloads.

Both GLB and power proportionality have been extensively studied in various contexts (e.g., electricity cost [23], [29], [30], [40], carbon emission [20]). Nonetheless, what makes our research unique is that we *exploit the inherent spatial and temporal diversities of data centers' water efficiencies and optimize GLB and power proportionality decisions for water conservation*.

It is a very *challenging* problem to minimize operational cost while capping water consumption using GLB and power proportionality. First, dispatching more workloads to water-efficient data centers may result in high electricity cost, and hence it is non-trivial to successfully cap water footprint while keeping the operational cost low. Second, when using power proportionality, incorrectly turning off too many servers may result in intolerable performance degradation, whereas turning off too few servers may unnecessarily increase operational cost and waste water. Last but not least, achieving water capping involves *budgeting* water consumption over a long term and hence optimally doing so requires the knowledge of far future information (e.g., outside temperature, which affects water efficiency), but such offline information is practically challenging to obtain accurately.

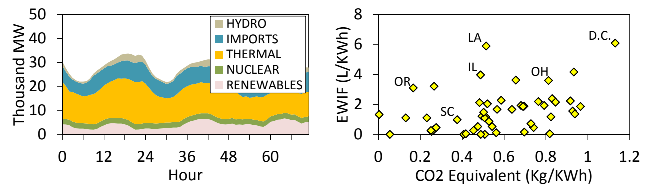
To address the above challenges, we propose a new online workload management algorithm, called WATCH (WATer-constrained workload sCHeduling in data centers), which can achieve a low operational cost while successfully capping the long-term water footprint without foreseeing the far future. The general intuition of WATCH is that if the actual water footprint has exceeded the expected level thus far, WATCH will put more emphasis on reducing the water consumption optimizing GLB and power proportionality decisions, such that the water deficit can be decreased and ultimately satisfy the desired water capping. We formally prove that WATCH results in a close-to-minimum operational cost compared to the optimal algorithm with complete offline information, given *arbitrary* run-time system dynamics. To evaluate WATCH under realistic settings,

1. Unless otherwise stated, "long-term" refers to one year and matches the current water accounting period [6], [45].



(a) (b)

Fig. 1. **(a)** Direct WUE versus outside wet bulb temperature [44]. Lines/markers represent modeled/measured values. **(b)** 3-day outside temperature starting from July 1, 2012 [11].



(a) (b)

Fig. 2. **(a)** 3-day fuel mix data of California ISO starting from Sep 1, 2013 [1]. **(b)** State-level EWIF versus CO_2 emissions in the U.S. [45], [47].

we perform a trace-based simulation, demonstrating that WATCH can significantly reduce the water footprint (e.g., by 20%) while incurring a small cost increase even compared to state-of-the-art cost-minimizing but water-oblivious solution. Finally, we extend WATCH in two different directions: (1) integrating WATCH with carbon footprint capping (which is a key aspect of sustainability); and (2) capping onsite cooling water consumption for each data center.

To sum up, our approach relies on yet optimizes the widely-available GLB and power proportionality knobs for long-term water capping. It has the following key features: **(1)** it requires little modification to the current software stacks and can be easily implemented, as there already exist various mechanisms to enable GLB and power proportionality [20], [23], [40]; **(2)** it is an online approach and does not require any long-term offline information; and **(3)** it only incurs a slight operational cost increase while slashing water consumption (e.g., by 20%, which is the water reduction percentage being urged in California [38]). While we recognize the importance of the existing water-saving techniques based on improved "engineering" [7], [10], [22], we emphasize that our proposed approach provides a complementary yet unique perspective to the current research. To our best knowledge, this is the first study to use workload management for long-term water capping in geo-distributed data centers.

2 Background

Following the metric developed by the Green Grid [45], we measure data center water efficiency as Water Usage Effectiveness (WUE), which is defined as the ratio of total water consumption to the IT energy usage. Instead of water withdrawal, we focus on water consumption (e.g., evaporation into the air), because it is a more accurate indicator of how much water does not return to source (i.e., "lost") [32].

- **Direct WUE.** As shown in a recent survey [46], cooling towers are commonly employed in large data centers as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

the heat rejection mechanism. In general, onsite (or direct) water consumption in cooling towers consists of two major parts: water evaporation and water "blown down", where the former is employed to transfer heat to the environment and the latter is for keeping salt concentration of the condenser water from becoming too high [16]. "Blown down" water consumption depends primarily on water quality: the higher water quality, the more cycles of concentrations (i.e., water recirculation times) and hence the less blown-down water [16]. Quantitatively, blown-down water can be expressed as one $(S-1)$-th of the evaporated water, where S is the cycle of concentrations (typically 3–10).

To highlight the impact of outside wet bulb temperature on the direct WUE, we present an empirical measurement model based on an industry cooling tower [44]. Specifically, following recommended operational settings, we show the direct WUE at different cycles of concentrations (i.e., water recirculation times) in Fig. 1(a), which clearly demonstrates that the direct WUE increases with outside wet bulb temperature (because at a lower wet bulb temperature, water cools down more by the outside air and less through evaporation). Using data fitting based on least square errors, we obtain an empirical direct WUE model as $\text{WUE}_{\text{direct}} = \frac{S}{S-1}\left(6\times10^{-5}\cdot T_{\text{w}}^3 - 0.01\cdot T_{\text{w}}^2 + 0.61\cdot T_{\text{w}} - 10.40\right)$, where S is the cycle of concentrations and T_{w} is the outside wet bulb temperature (in Fahrenheit).

• **Indirect WUE.** We now present the indirect water efficiency following [20], [25]. Indirect WUE depends on the energy fuel mixes (e.g., coal, nuclear, hydro) as well as cooling techniques used by power plants and hence is also called Electricity Water Intensity Factor (EWIF) [32], [45]. Since electricity produced by different energy fuels becomes non-differentiated once entering the grid, we consider the average EWIF which can be estimated as $\overline{EWIF} = \frac{\sum_k b_k \times EWIF_k}{\sum_k b_k}$, where b_k denotes the amount of electricity generated from fuel type k in location serving the data center, and $EWIF_k$ is the EWIF for fuel type k [20], [25]. Note that variations in energy fuel mixes of electricity generation (to meet various demand levels, shown in Fig. 2(a)) results in temporal diversity of EWIF. Further, indirect EWIF also varies by location, because each fuel type has its own distinct EWIF [45] and energy fuel mix is typically different between states as some states may use less water-efficient energy generation than others [45].

Finally, we show in Fig. 2(b) the state-level EWIF versus average carbon emission rate in 50 U.S. states: "greener" states may not necessarily be less "thirstier" [32], [45].[2] This implies that the existing research (e.g., [20]) that favors carbon reduction may result in more water footprint. Similar statements also hold for water efficiency versus electricity price, whose details are omitted for brevity. As a result, the existing cost-/carbon-driven techniques do not necessarily lead to water conservation, thereby necessitating the exploitation of spatio-temporal diversities of water efficiency to survive drought.

2. Readers may refer to [45] for detailed EWIF data, and to [47] for complete carbon efficiency data.

TABLE 1
List of key notations.

Notation	Description
M_i	Total no. of servers in data center i
m_i	No. of servers turned ON at data center i
a_i	Workload dispatched to data center i
λ_j	Workload arriving at gateway j
μ_i	Service rate of a server in data center i
γ_i	PUE of data center i
$l_{i,j}$	Network delay from gateway j to data center i
e_i	Electricity cost of data center i
w_i	Water footprint of data center i
Z	Long-term water footprint constraint
g	Total cost
q	Water budget deficit queue

3 Water-Aware Workload Management

In this section, we present a water-aware online workload management algorithm that can successfully cap the overall water footprint under a desired level for data centers' survival of drought while incurring a negligible penalty in other aspects such as operational cost. Towards this end, we describe the general approach, present the model, formulate the problem and then present the algorithm WATCH.

3.1 General approach

Our proposed research for data centers' survival of drought relies on the following two technical approaches.

• **Geographic load balancing:** Many large IT companies are operating data centers in geographic locations for redundancy/latency concerns. Incoming workloads, especially "request-response" web services (e.g., search, e-commerce), can be flexibly scheduled among multiple data centers using HTTP redirection or persistent HTTP proxies to tunnel requests [33].

• **Power proportionality:** The basic principal to enable power proportionality (also referred to as "dynamic right-sizing") is to turn off unused servers, as static/idle servers may consume even 60% of full power [29]. Hence, turning off some servers when workload is low can effectively reduce energy consumption as well as water footprint.

While both GLB and power proportionality have been extensively studied [19], [23], [27], [29], [30], we focus on *optimizing* GLB and power proportionality decisions for data center's water footprint capping: **(1)** how to dispatch workloads to geo-distributed data centers (i.e., the percentage of incoming workloads scheduled to each data center); and **(2)** how many servers to turn on in each data center.

3.2 Model

We consider a discrete-time decision model by dividing the entire time period of interest (e.g., typically a year, as used by LEED [48] and suggested by [45]) into K time slots. For example, each time slot can be one hour [30].

Data center. We consider N geo-distributed data centers, indexed by $i = 1, 2, \cdots, N$. The service capacity of data center i is represented by M_i homogeneous servers each having a service rate of μ_i (i.e., the *average* number of jobs that can be processed in a unit time). We denote by $m_i(t)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

the number of servers turned on in data center i. We model data center power based on the utilization as follows

$$p_i(a_i(t), m_i(t)) = \gamma_i(t) \cdot m_i(t) \cdot \left[e_{0,i} + e_{c,i} \frac{a_i(t)}{m_i(t)\mu_i} \right],$$

where $\gamma_i(t)$ is the Power Usage Effectiveness (PUE) factor, $a_i(t) = \sum_{j=1}^{J} \lambda_{i,j}(t)$ is the total amount of workloads dispatched to data center i (with $\lambda_{i,j}(t)$ being the amount of workloads originating from the j-th gateway, as will be specified later), $e_{0,i}$ is the static server power regardless of the workloads (as long as a server is turned on) and $e_{c,i}$ is the dynamic power when a server is busy. Essentially, $\frac{a_i(t)}{m_i(t)\mu_i}$ is the server utilization. This model can capture server power with a reasonable accuracy, validated by real-world measurements [17] and extensively used in prior studies [29], [30], [41].

Electricity cost. As in [30], [40], [41], we consider real-time pricing (due to electricity market deregulation) and denote the electricity price in data center i at time t by $u_i(t)$. Hence, the incurred electricity cost of data center i is

$$e_i(a_i(t), m_i(t)) = u_i(t) \cdot p_i(a_i(t), m_i(t)). \quad (1)$$

Water consumption: The direct cooling water consumption can be obtained by multiplying server power consumption with the direct WUE, while the indirect water consumption depends on the electricity usage as well as the local EWIF. Thus, we can express the water consumption of data center i at time t as

$$w_i(t) = \left[\frac{\epsilon_{i,D}(t)}{\gamma_i(t)} + \epsilon_{i,I}(t) \right] \cdot p_i(a_i(t), m_i(t)), \quad (2)$$

where $\epsilon_{i,D}(t)$ is the direct WUE (i.e., ratio of water to IT energy) at time t and $\epsilon_{i,I}(t)$ is the EWIF (i.e., ratio of water to electricity production) of the electricity powering data center i.

Workload. As in many of the prior GLB studies [20], [30], we focus on delay-sensitive interactive workloads (also interchangeably referred to as jobs) that can be flexibly scheduled among multiple data centers. There are J gateways, each of which represents a geographically-concentrated source of workloads (e.g., a state or province). We denote the workload arrival rate at the j-th gateway by $\lambda_j(t) = [0, \lambda_{j,\max}]$, and the workload is dispatched to data center i at a rate of $\lambda_{i,j}(t)$ that we shall optimize. Given a certain number of servers, the delay performance intuitively becomes worse with more dispatched workloads. Here, we consider the *overall* delay performance by modeling the service process at each server as an M/M/1 queue [29], [30]. Specifically, the total delay in data center i can be written as

$$d_i(a_i, m_i) = \sum_{j=1}^{J} \lambda_{i,j} \cdot \left[\frac{1}{\mu_i - \frac{a_i}{m_i}} + l_{i,j} - d_{th} \right]^+, \quad (3)$$

where the operator $[\ \cdot\ ]^+ = \max\{0, \cdot\}$ indicates that the revenue/profit loss is negligible when the average delay is smaller than the threshold d_{th} (i.e., little user experience improvement when the delay is already sufficiently small), and $l_{i,j}$ is average network delay approximated in proportion to the distance between data center i and the j-th gateway [30].

3.3 Problem formulation

We focus on *operational* cost rather than capital cost (e.g., building data centers), which can be minimized using separate techniques [42]. Two types of "costs" are considered: electricity cost and delay "cost", where the former takes up a dominant fraction of the operational cost while the later affects user experiences and revenues [28], [30]. We will investigate the bandwidth cost in the next section. As water bill has yet to catch up with electricity cost and indirect water consumption is "paid" in energy bills, we do not incorporate water cost in our work. As in the literature [20], [28], [30], our study considers the following parameterized cost

$$g(\lambda(t), \mathbf{m}(t)) = \sum_{i=1}^{N} \left[e_i(a_i(t), m_i(t)) + \beta \cdot d_i(a_i(t), m_i(t)) \right],$$

where $\lambda(t) = (\lambda_{1,1}(t), \cdots, \lambda_{1,J}(t), \cdots, \lambda_{N,1}(t), \cdots, \lambda_{N,J}(t))$ and $\mathbf{m}(t) = (m_1(t), \cdots, m_N(t))$ represent GLB and power proportionality decisions, respectively, and $\beta \geq 0$ is the weight parameter converting delay performance to monetary "cost" (adjusting tradeoff between electricity cost and delay performance) [28], [30]. Throughout the paper, we use "operational cost" to represent the parameterized cost. Next, we formulate the problem as follows

$$\mathbf{P1}: \quad \min_{\mathcal{A}} \bar{g} = \frac{1}{K} \sum_{t=0}^{K-1} g\left(\lambda(t), \mathbf{m}(t)\right) \quad (4)$$

$$s.t., \quad 0 \leq \sum_{j=1}^{J} \lambda_{i,j}(t) \leq \eta \cdot \mu_i \cdot m_i(t),\ \forall i, t, \quad (5)$$

$$m_i(t) \leq M_i,\ \forall i, t, \quad (6)$$

$$\sum_{i=1}^{N} \lambda_{i,j}(t) = \lambda_j(t),\ \forall j, t, \quad (7)$$

$$\sum_{t=0}^{K-1} \sum_{i=1}^{N} w_i(t) \leq Z, \quad (8)$$

where $\mathcal{A}$ represents a sequence of GLB and power proportionality decisions, i.e., $\lambda(t)$ and $\mathbf{m}(t)$, for $t = 0, 1, \cdots, K-1$. The constraints (5), (6) and (7) indicate no server overloading, over-provisioning or workload dropping, while the constraint (8) specifies the long-term water consumption constraint. In constraint (5), $\eta \in (0, 1)$ specifies the maximum server utilization (equivalently, the worst delay performance). Note that additional constraints, such as that some workloads may only be routed to certain data centers, can also be incorporated into our study. Moreover, we will also consider onsite water capping for each data center (which is more related to regional drought) and carbon footprint capping (another sustainability criterion [20]) in Section 5.

In **P1**, the power usage is an affine function of load distribution $\lambda_{i,j}(t)$ and the number of servers to turn on $m_i(t)$, and so are electricity cost and water footprint. The delay cost is a convex function of $\lambda_{i,j}(t)$ and $m_i(t)$. Thus, **P1** is convex optimization which can be solved in polynomial time [12].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

3.4 WATCH

In this subsection, we develop our water-aware workload management algorithm WATCH which can be implemented online without foreseeing offline information.

Main Challenge. Addressing data centers' water footprint in the face of extended droughts requires long-term efforts, but the long-term nature also creates challenges as the desired water capping constraint in (9) couples the workload management decisions across different time slots: while GLB and power proportionality decisions have to be made without foreseeing the far future, the current decisions will implicitly affect the future decisions (e.g., using more water at this time slot will result in less water budget available for future uses). Accurate prediction of such offline information (e.g., volatile outside wet bulb temperature which affects WUE, non-stationary workload arrival, etc.) is quite challenging, if not impossible [19], necessitating an online approach.

Solution. To address the lack of long-term future information, we leverage the sample-path Lyapunov technique [37] to develop an online algorithm that makes GLB and power proportionality decisions only based on currently available information. Originally proposed for establishing control system stability, Lyapunov technique was later extended to achieve long-term queueing stability in networks [37], with a salient feature that it does not require future information when making control decisions. Here, we can treat the data center water footprint in each time slot as "job arrivals" to a *virtual* water queue, and view the desired water usage as "job departures". Thus, if the virtual queue length can be pushed to zero at the end, then the desired long-term water footprint capping is achieved in an online manner.

At the core of our solution is the (virtual) water budget deficit queue which *replaces* the long-term constraint (9) and, with an initially empty queue $q(0) = 0$, evolves as follows

$$q(t+1) = \left\{ q(t) + \sum_{i=1}^{N} w_i(t) - z(t) \right\}^{+}, \tag{9}$$

where $q(t)$ is queue length at beginning of time slot t, $\sum_{i=1}^{N} w_i(t)$ is the combined water consumption of all the data centers and $z(t)$ is the reference water budget for time slot t. The reference budget $z(t)$ can be a constant (e.g., $z(t) = Z/K$ for $t = \{0, 1, \cdots K-1\}$) or estimated based on projected workload (which does not need to be accurate) such that $\sum_{t=0}^{K-1} z(t) = Z$. However, based on simulation studies (not included in this paper for brevity), we note that the choice of $z(t)$ has a negligible impact on the outcome of WATCH in terms of the cost efficiency, provided that the the total water budget is the same. This is partly due to the fact that $z(t)$ is only a reference value that places no enforcement on the execution of WATCH.

The queue length at any time indicates the deviation of actual water usage from the total reference water usage, and we integrate this information (i.e., $q(t)$) in our optimization so that WATCH can act on the deviation to mitigate it. Specifically, instead of optimizing the original objective function, we construct in WATCH a new objective function, $V \cdot g(\lambda(t), \mathbf{m}(t)) + q(t) \cdot \sum_{i=1}^{N} w_i(t)$, which is the original objective function scaled by a parameter V plus the water usage multiplied by the water budget deficit queue $q(t)$ in (9). Thus, there is no decision coupling across different time slots in the new problem, and hence it can be solved *online* requiring only the information (e.g., workload, WUE etc.) of current time slot.

Algorithm 1 WATCH

1: Input $\lambda_j(t)$, $\epsilon_{i,D}(t)$, $\epsilon_{i,I}(t)$, and $u_i(t)$ at the beginning of each time $t = 0, 1, \cdots, K-1$, for $i = 1, 2, \cdots, N$

2: Choose $\lambda(t)$ and $\mathbf{m}(t)$ subject to constraints (5),(6),(7) to minimize

$$\mathbf{P2}: \quad V \cdot g(\lambda(t), \mathbf{m}(t)) + q(t) \cdot \sum_{i=1}^{N} w_i(t) \tag{10}$$

3: Update $q(t)$ according to (9).

The water budget deficit queue acts as a feedback from past decisions to the current decision, so that deviation from the long-term water footprint constraint can be mitigated. The new objective function $V \cdot g(\lambda(t), \mathbf{m}(t)) + q(t) \cdot \sum_{i=1}^{N} w_i(t)$ is devised such a way that the mitigation takes place gradually so that the feedback does not overshadow the main objective of cost minimization. In particular, the parameter V acts as a control parameter that determines how much emphasis to put on cost minimization compared to the long-term water footprint constraint. Larger V implies that the data center cares more about cost minimization than meeting the long-term water budget, and vice versa. Naturally, when cost minimization is prioritized, long-term water consumption constraint is more likely to be exceeded with a larger gap, which is also substantiated in analytical study of WATCH presented in Theorem 1 as well as demonstrated through our simulation results.

Algorithm input/output. Algorithm 1 only requires online information as specified in Line 1, which can be readily obtained: e.g., outside wet bulb temperature can be measured online, whereas workload arrival rate can be well estimated using various learning algorithms [30]. We will also demonstrate in the next section that WATCH is robust against inaccurate online information. At the beginning of each time slot, Algorithm 1 outputs the GLB decision $\lambda(t)$ (i.e., the portion of workloads dispatched to each data center) and power proportionality decision $\mathbf{m}(\mathbf{t})$ (i.e., how many servers are turned on in each data center).

Analysis. The following theorem formally shows the performance of WATCH.

Theorem 1. *For any $T \in \mathbb{Z}^+$ and $H \in \mathbb{Z}^+$ such that $K = HT$, the following statements hold.*

a. *The water consumption constraint is* approximately *satisfied with a bounded deviation:*

$$\sum_{t=0}^{K-1} \sum_{i=1}^{N} w_i(t) \le Z + \sqrt{KC(T) + \frac{KV}{H} \sum_{h=0}^{H-1} (G_h^* - g_{\min})}, \tag{11}$$

where $C(T) = U + D(T-1)$ with U and D being certain finite constants satisfying $U \ge \frac{1}{2} \max_{t=0,1,\cdots,K-1} [\sum_{i=1}^{N} w_i(t) - z(t)]^2$ and $D = \frac{1}{2} \left[\max_{t=0,1,\cdots,K-1} \{\sum_{i=1}^{N} w_i(t), z(t)\} \right]^2$, G_h^* is the minimum average operational cost by the optimal offline

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

algorithm with T-slot lookahead information over $t = (h-1)T, \cdots, hT-1$, for $h = 0, 1, \cdots, H-1$, and $g_{\min}$ is the minimum possible operational cost.

b. *The average operational cost $\bar{g}^*$ achieved by WATCH satisfies:*

$$\bar{g}^* \leq \frac{1}{H}\sum_{h=0}^{H-1} G_h^* + \frac{C(T)}{V}. \tag{12}$$

Proof. As a sketch, we only outline the key steps of proving Theorem 1, while the proof details can be established following Lyapunov-drift-plus-cost technique [37]. Note first that G_h^* is the minimum cost achieved by the offline algorithm with T-step lookahead information by solving: $\min_{\mathcal{A}} G_h^* = \frac{1}{T}\sum_{t=hT}^{(h+1)T-1} g(\lambda(t), \mathbf{m}(t))$ subject to (5)(6)(7) and "$\sum_{t=hT}^{(h+1)T-1}\sum_{i=1}^{N} w_i(t) \leq \sum_{t=hT}^{(h+1)T-1} z(t)$". We need to define a quadratic Lyapunov function $L(q(t)) \triangleq \frac{1}{2}q^2(t)$, and then derive a finite upper bound on the T-step Lyapunov drift, i.e., difference between $L(q(t+T))$ and $L(q(t))$. By adding operational cost into both Lyapunov drift and upper bound, we will see that **P2** in Algorithm 1 is essentially minimizing the upper bound on Lyapunov drift plus operational cost. Then, after mathematical manipulations, Theorem 1 will follow. Unlike prior work [24], [52], we use sample-path techniques without making specific probability, i.i.d./Markovian or even "slater" assumptions [37] over system dynamics and hence, Theorem 1 holds under arbitrary dynamics. ■

Theorem 1 shows that, given a fixed value of T and H, WATCH can approximately satisfy the long-term water footprint constraint with a bounded deviation, whereas the cost of WATCH is within an additive penalty compared to the minimum cost achieved by the offline algorithm with T-step lookahead information (which solves $\min_{\mathcal{A}} G_h^* = \frac{1}{T}\sum_{t=hT}^{(h+1)T-1} g(\lambda(t), \mathbf{m}(t))$ subject to (5)(6)(7) and "$\sum_{t=hT}^{(h+1)T-1}\sum_{i=1}^{N} w_i(t) \leq \sum_{t=hT}^{(h+1)T-1} z(t)$"). From (11) and (12) we see that when V increases cost performance of WATCH is closer to the offline algorithm with lookahead information while the potential deviation from long-term water consumption target becomes larger, and vice versa. Hence, WATCH presents with an online treatment for **P1** with an analytical bound on how far WATCH can be from the optimal solution (with T-step lookahead information). The analytical observation on WATCH's performance is corroborated by our real-life trace-based simulation studies.

4 Performance Evaluation

This section presents trace-based simulation studies to validate our analysis. We first present our simulation setup and then show that WATCH can significantly reduce the water consumption compared to the existing cost-minimizing GLB algorithm (by 20%) while only incurring a small increase in the operational cost (even compared to state-of-the-art cost minimizing algorithm). We also show the benefits and robustness of WATCH under different settings.

4.1 Setup

Considering the practical difficulty in implementing WATCH in real systems, we resort to simulations following the common practice [20], [30], [40].

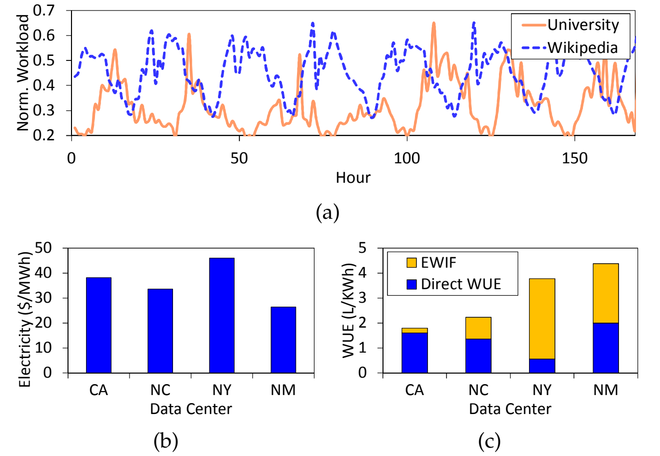


Fig. 3. Simulation setup. **(a)** Wikipedia [49] and university workloads. **(b)** Average electricity price [47]. **(c)** Average WUE.

We consider four geographically distributed data centers located in: (#1) Mountain View, CA, (#2) Forest City, NC, (#3) New York, NY, and (#4) Roswell, NM. The number of servers in these data centers are: 60K, 50K, 40K, 25K, respectively. To limit the number of free parameters, the default PUEs for all data centers are chosen as 1.20, although PUEs may vary over time [10]. While different data centers typically have different server configurations, we focus on homogeneous servers and each server has a normalized service rate of 1.00 (i.e., one *unit* of workloads per second). Each server has a maximum power of 400W, and static/idle server power takes up 60% of the maximum power. As in the existing work [20], [28], we consider delay-sensitive workloads. The weight converting the delay performance to cost is set to $\beta = 12$, although WATCH is applicable for any settings. All the workloads are distributed to the four data centers by one front-end gateway located in North Platte, NE, which has comparable distances to all data centers. GLB and power proportionality decisions are updated hourly. Accordingly, all the operational cost and water consumption are hourly values unless otherwise stated.

The time horizon is one year. The average water consumption by state-of-the-art cost-minimizing GLB algorithm (presented in [30], which disregards water footprint) is chosen as our reference value, and in our setting using default workload traces, it is 177 KL (kilo-liters) per hour. We choose 142 KL per hour on average (or equivalently, around 332 million gallons per year) as the default capping constraint, which is 80% of the reference value. This 20% water reduction also matches the target set by California [2]. We also use Low (L), Medium (M), and High (H) water capping to represent 75%, 80% and 85% of reference water consumption.

• Electricity price: We obtain from [47] hourly wholesale electricity prices from four trading nodes closest to our considered data centers for the year of 2012. The average electricity prices in our simulations are shown in Fig. 3(b).

• EWIF and direct WUE: Due to the lack of access to EWIF data in our data center locations, we use the state-level average EWIF values calculated based on the data in [32], [45]. For direct WUE, we use the empirical values in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

[44] modeled as a function of outside wet bulb temperature (see Fig. 1(a)). The temperature data for the year of 2012 is obtained from [11]. Different data centers may use different cooling techniques/towers [5], [10]. Nonetheless, only Facebook is disclosing its real-time water efficiency information [10]. Thus, to reflect geographic diversities of direct WUE caused by non-weather factors, we choose to scale the direct WUE (obtained from our empirical model) differently for these four data centers. Average direct WUEs and EWIFs are both shown in Fig. 3(c).

• Workloads: As our default workload, we scale the Wikipedia workload trace [49] and extend it to one year by adding up to 30% random noises. We also obtain the workload trace by profiling the server usage log of a large public university from January 1 to December 31, 2012, and scale up the arrival rate. The workload trace contains request-level records, such as arrival time, completion time, size of data sent, service status for each request. The normalized workload arrival rates are shown in Fig. 3(a), where the values are normalized with respect to the maximum capacity of all data centers.

Remark. As our research takes an early step to address data center operation in drought conditions and due to lack of publicly available data (especially real-time WUE), we obtain traces and infer data from various sources. Admittedly, if all the data is available to us from production systems, the specific experimental results may differ, but we expect that the general trend still holds, as the advantage and intuition of WATCH have been demonstrated both conceptually and analytically in previous sections. Thus, we do not intend to emphasize our quantitative results in simulations, but rather we would like to leverage simulations to provide additional justification to WATCH under reasonably realistic settings.

4.2 Results

In this subsection, we present simulation results using the above trace data.

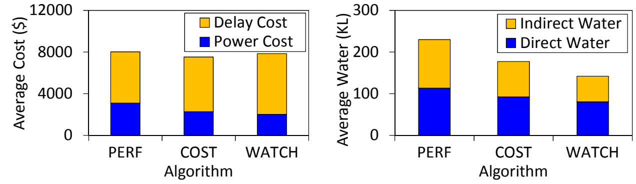
4.2.1 *Performance comparison*

We first present two widely-studied benchmark algorithms and then subsequently compare WATCH against them.

• **COST:** While it is not possible to compare WATCH against all existing techniques, we choose cost-driven but water-*oblivious* algorithm that minimizes the operational cost [30], referred to as COST, as our benchmark, since it is *one* of the most widely-considered benchmarks and our objective is also minimizing cost.

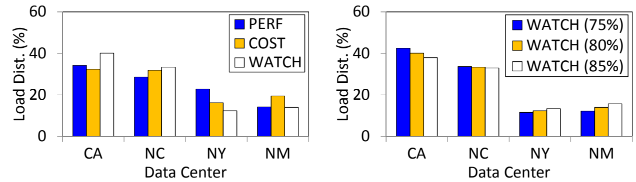
• **PERFORMANCE** (abbreviated as "PERF" to save space): The current practice is still performance-driven: turning on all the servers and scheduling requests to the nearest data center if applicable. Hence, we consider this approach as the second benchmark and refer it to as PERF.

Comparable operational cost. Fig. 4(a) shows the average hourly operational costs (incorporating both electricity cost and delay performance) achieved by PERF, COST and WATCH (with cost-water parameter $V = 45$). While WATCH aims at minimizing operational cost, it also considers water consumption as its additional constraint. Thus, as can be seen from Fig. 4(a), WATCH incurs a slightly higher operational cost (by approximately 3%) compared to COST



(a) (b)

Fig. 4. Comparison among WATCH, COST, and non-GLB. **(a)** Operational cost. **(b)** Water consumption.



(a) (b)

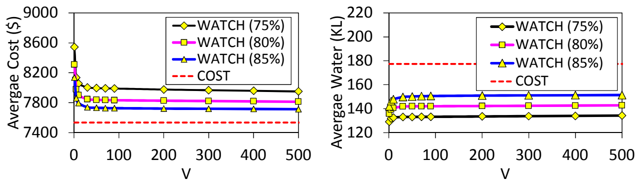
Fig. 5. Workload distributions. **(a)** Different algorithms. **(b)** WATCH under different water caps.

which explicitly minimizes the operational cost: WATCH and COST are comparable in both electricity cost and delay performance. As water-efficient data centers are typically different from cost-effective ones (as discussed in Section 2 and can be seen from Fig. 3(b) and Fig. 3(c)), it may not be possible to optimize workload management decisions for both metrics *simultaneously*: an inherent tradeoff exists between water consumption and operational cost. While PERF incurs higher energy consumption, it provides the best performance. Hence, after we convert the delay into cost, the total operational cost of PERF is almost the same as WATCH, but it is most water-consuming and hence vulnerable to drought conditions. Note also that, although not shown for brevity, WATCH can reduce electricity usage compared to COST by about 10% in our case study .

Reduced water consumption. WATCH explicitly incorporates spatio-temporal diversities of water efficiency into its workload management decisions, and uses water deficit queue as a guidance towards water capping. Fig. 4(b) demonstrates that WATCH can successfully meet the water consumption constraint which, by default, is only 80% of the water consumption by COST. Combined with Fig. 4(a), we see that WATCH can lead to 20% saving of water footprints while incurring nearly no operational cost. Expectedly, if we set a less aggressive water conservation target (e.g., 15% saving), WATCH will result in a lower cost.

Water-driven scheduling. Next, we show in Fig. 5 the *average* workload distributions across different data centers. The performance-driven PERF distributes workloads in proportion to data center server capacity, offering the same delay performance across data centers. As COST focuses on minimizing the operational cost, it favors the data center in NM with the lowest average electricity price: COST distributes more workloads to the cost-effective NM data center than to NY, even though the former has a less capacity than the latter. Now, we turn to Fig. 5(b) to show average workload distributions for WATCH under different water caps (i.e., Low, Medium, and High). By looking at Fig. 3(c),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing



(a) (b)

Fig. 6. Cost-water tradeoff under different water caps. **(a)** Operational cost. **(b)** Water consumption.

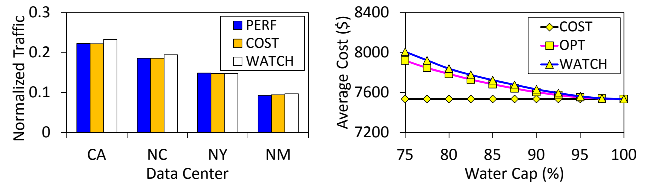
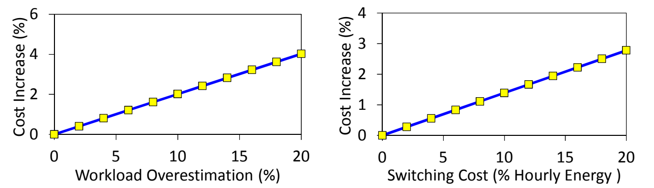


Fig. 7. **(a)** Traffic assignment comparison. **(b)** Impact of water capping constraints on operational cost.



(a) (b)

Fig. 8. Sensitivity study. **(a)** Workload prediction error. **(b)** Server switching cost.

we see that the NM data center has the highest average WUE, whereas the CA data center has the lowest one. Thus, WATCH will schedule more workloads to CA when water consumption constraint is more stringent (i.e., Low water budget), while more workloads will be diverted to NM for cost efficiency when water consumption constraint is less stringent (i.e., High water budget), although NM has the highest average WUE.

4.2.2 *Cost versus water consumption tradeoff*

As we have formally proved in Theorem 1, the cost-water parameter V governs the flexible tradeoff between cost minimization and water consumption: the larger V, the more emphasis on reducing the cost while potentially deviating more from the desired water footprint constraint, and vice versa. To illustrate this point, we show in Fig. 6(a) and Fig. 6(b) the impact of V on the average hourly cost and water consumption, respectively, under different water caps. The result conforms with our analysis that with a greater V, WATCH is less concerned with water consumption while caring more about the cost. In the extreme case in which V goes to infinity, WATCH reduces to a *water-oblivious* algorithm (i.e., COST) which minimizes the operational cost without considering water consumption. Clearly, water-oblivious COST algorithm achieves a cost that is a lower bound on the cost that can be possibly achieved by any algorithm satisfying the water cap, but it is not desirable in water-stressed areas.

4.2.3 *Bandwidth cost comparison*

While bandwidth cost has been shrinking in recent years relative to electricity cost [40], it may still be a non-negligible portion of data center operational cost. As WATCH may change the traffic patterns among data centers, the bandwidth cost may change as well. In this paper, we focus on the prevailing 95/5 bandwidth charging model: the 95th percentile of data center traffic, measured in 5-minute intervals, is used for billing [20], [40]. In our study, we consider the link traffic between the gateway and the data center is proportional to the assigned workload (because the average data size of a job request is relatively constant). We then measure the workloads assigned to each data center for every 5 minutes and take the 95-percentile traffic during each month. Finally, we average the results to get the yearly average bandwidth cost for different algorithms under the default 20% water reduction target. Fig. 7(a) shows the normalized 95/5 traffic assignment (normalized with respect to the maximum capacity of all data centers). We see that the 95/5 traffic assignments by WATCH, PERF and COST are almost the same. Although the data centers in CA and NC have slightly higher 95/5 traffic by using WATCH than using PERF or COST, the difference in the resulting bandwidth cost is almost negligible (within 5%). Considering the decreasing trend of bandwidth cost but surge in energy cost, we see that WATCH is still attractive in terms of slashing water while potentially incurring a small bandwidth cost increase.

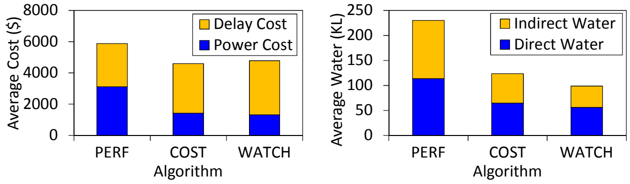
4.2.4 *Sensitivity study*

We now perform various sensitivity studies, with V chosen such that water capping constraint is satisfied.

• Water capping constraint: We now show in Fig. 7(b) the impact of water capping constraint (i.e., water budget) on the operational cost. We normalize the water budget with respect to the water consumption by COST (i.e., average 177KL per hour). It is seen that given a 80% water budget — equivalent to using only 80% of the water consumed by COST, WATCH exceeds the water-oblivious COST algorithm by approximately 3% in terms of the average operational cost, while still being able to satisfy water consumption constraint (which is clearly violated by COST). More interestingly, when the normalized water budget increases to 95%, WATCH achieves almost the same cost as COST (within 1%). Hence, by exploiting spatio-temporal diversities of water efficiency, WATCH can conserve water by 5% almost for "free" compared to COST. As a further comparison, we also show the cost of the optimal offline algorithm, referred to as OPT, in Fig. 7(b), and it can be seen that WATCH is quite close to OPT in terms of the cost, demonstrating that WATCH performs remarkably well by only using online information.

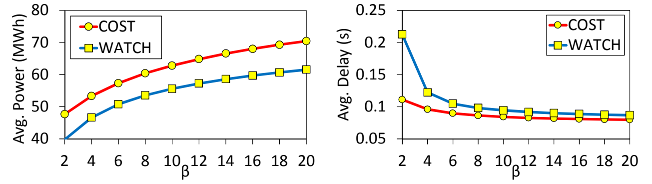
• Workload overestimation: In practice, it may not be possible to perfectly estimate the current workload arrival rate. To handle possible traffic spikes, data centers may leave a capacity margin by turning on more servers than needed as a backup or deliberately overestimate the workload arrival rate by a certain overestimation factor $\phi \geq 1$: the higher ϕ, the more overestimates. We choose the later approach, and scale up workload by ϕ during optimization to decide

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing



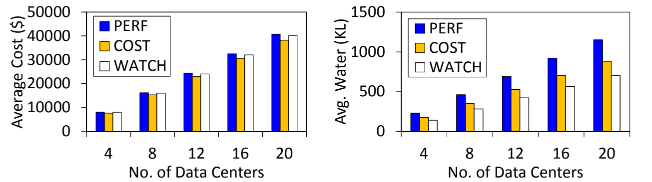
(a) (b)

Fig. 9. Comparison among WATCH, COST, and non-GLB (University workload). **(a)** Operational cost. **(b)** Water consumption.



(a) (b)

Fig. 10. Impact of β.



(a) (b)

Fig. 11. Performance comparison for many data centers.

the optimum number of servers to be turned on in each data center and the load distribution. However, when we determine the cost, power, delay and water consumption, we still use the actual workload. We keep all the other parameters (such as β, $z(t)$, the electricity price, etc.) as unchanged. Fig. 8(a) shows that the total operation cost only increases by around 4%, even when we overestimate the workloads by 20% (which is already sufficiently high in practice, as shown in [19]). This is because although workload overestimation may turn on more servers than needed and incur a higher electricity cost, it has a lower delay cost as the delay performance is improved because of increased number of servers.

• Switching cost: Switching servers on/off induces various costs, such as energy/time waste as well as “wear and tear”. As in [28], we incorporate all these factors and use switching cost as the combined cost quantified in terms of *energy* consumption. We normalize the switching cost (incurred by turning on/off *one* server) with respect to the maximum hourly energy consumption of a single server. Fig. 8(b) shows that even when the switching cost of one server takes 20% of its maximum hourly energy consumption, the average operational cost only increases by less than 3% while satisfying water capping.

• Different workloads: Now, we use the university workload trace to drive our simulations for demonstrating the applicability of WATCH under various workloads. The results are shown in Fig. 9. As in Fig. 4, it can be seen that the same message can be delivered in Fig. 9: WATCH achieves a cost fairly close to that of water-oblivious COST, and meanwhile significantly slashes the water consumption.

• Impact of β: The parameter β determines the relative weight of delay performance as compared to electricity cost. To show the impact of β parameter on data center's electricity cost and resulting delay performance, we vary the value of β and show the corresponding results in Fig. 10. The water budget for each β is set to 80% of the corresponding water usage by COST. Thus, when COST increases its own water usage, the water budget for WATCH also increases. We see that the average delay decreases and average electricity cost increases as we increase β. In our study, we set $\beta = 12$ to have an average delay cost that is comparable to the average power cost. Moreover, with our choice of $\beta = 12$, the delay performance of WATCH is fairly close to COST. As shown in Fig. 10(b), setting a larger β in our simulation settings will significantly increase the electricity cost but only yield a very small decrease in delay (in tens of millisecond range), which is insignificant for human perception. Note that specific value of β also depends on the unit used for measurement as well. For example, we set $\beta = 12$ as we use “second” as the unit for delay, whereas if we use “millisecond” we would get the same results for $\beta = 0.012$.

• Number of data centers: In our study, we consider four data center locations by default, which is reasonable considering that even leading IT companies, like Facebook, only have a few self-managed megascale data centers (e.g., four data centers throughout the world, including two in the U.S. [9]). Nonetheless, we have extended our study to more data centers. We select additional locations for data centers and follow the same evaluation methodology as we have used for the default setting. Fig. 11(a) shows that WATCH still achieves an average cost that is very close to COST (cost-minimizing water-oblivious geographic load balancing). We also see in Fig.11(b) that regardless of the number of data centers, WATCH can meet the water footprint constraint, thereby translating water footprint reduction.

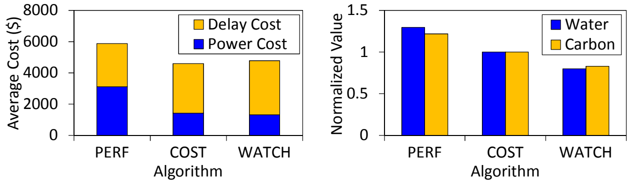
We also study WATCH's robustness against power, delay and water consumption modeling error/uncertainties, and see that at even at $\pm 20\%$ random modeling error, average cost only increases by 5% while the budget deficit still remains under 0.08% (given the same V). The supporting figures for this study are omitted for space limitation.

To sum up, it is not absolutely “free” to achieve water conservation due to the inherent tradeoff between water efficiency and cost efficiency. Nonetheless, WATCH exploits the spatio-temporal diversities of water efficiency and can cut water consumption by 20% while only incurring a small cost increase, even compared to the cost-minimizing algorithm COST. While we do not imply that WATCH outperforms the existing GLB techniques in all aspect, we emphasize that WATCH is complementary to the existing research and that it is particularly appealing for data center operation in water-stressed areas.

5 Extension

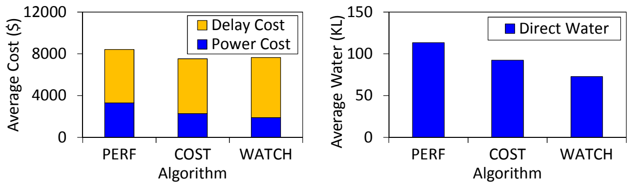
In this section, we extend WATCH in two directions: (1) capping carbon footprint; and (2) capping onsite cooling water for each data center.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing



(a) (b)

Fig. 12. Comparison among PERF, COST, and WATCH. **(a)** Operational cost. **(b)** Water consumption and carbon emission.



(a) (b)

Fig. 13. Comparison among PERF, COST, and WATCH. **(a)** Operational cost. **(b)** Total onsite water consumption.

5.1 Capping carbon footprint

It is widely accepted that carbon footprint is an important aspect of sustainability. Following [20], we express the carbon emission of data center i as

$$c_i(t) = h_i(t) \cdot p_i(a_i(t), m_i(t)), \tag{13}$$

where $p_i(a_i(t), m_i(t))$ is the electricity usage and $h_i(t)$ is the carbon emission rate calculated based on [20], with a unit of g/kWh, for power plant serving data center i. Compared to water footprints that are both onsite and offsite for data centers, carbon emission does not include the "*onsite*" component.

Like in Section 3.4, we incorporate another queue — carbon deficit queue — which evolves as $q_c(t+1) = [q_c(t) + \sum_{i=1}^{N} c_i(t) - z_c(t)]^+$, where $z_c(t)$ is the reference carbon budget for time t guiding GLB and power proportionality decisions towards carbon footprint capping. Then, we put an additional term $q_c(t) \cdot \sum_{i=1}^{N} c_i(t)$ into Line 2 of Algorithm 1. The results, including performance analysis, are similar. Fig. 12 demonstrates that, compared to cost-driven COST, WATCH can still successfully slash both water and carbon footprints while just incurring a small cost increase.

5.2 Capping onsite water consumption

As onsite cooling water is more directly related to data centers' water accounting [6] and drought condition is often region-specific [4], we extend WATCH to cap onsite water consumption for each data center. Specifically, instead of keeping track of only one water budget queue, we construct N water budget deficit queues, each representing the current onsite water deficit for one data center. Specifically, the water budget queue $q_i(t)$ for data center i evolves as $q_i(t+1) = [q_i(t) + w_i(t) - z_i(t)]^+$ and will be added into Line 2 of Algorithm 1. Fig. 13 demonstrates that WATCH can still reduce the onsite water consumption of each data center by 20% while keeping operational cost low.

6 Related Work

There has been a significant amount of research in optimizing data center operation from various perspectives [34], ranging from energy-aware task scheduling and resource allocation [51], [53], cutting electricity bills (using GLB and/or power proportionality) [23], [40], [41], minimizing response times [18], brown energy reduction [13], [30], carbon footprint minimization [15], [20], to addressing the thermal/reliability issues [35], [39]. Complimentary to our work, [14], [21] study energy efficiency of the network infrastructure for implementing GLB in data centers. Nonetheless, none of these studies have considered water consumption which is emerging as a critical concern for data centers' survival of drought.

Long-term optimization has been increasingly considered in the literature. For example, [42] explores the optimal energy portfolio for reducing carbon emissions, and [27] considers GLB with yearly energy capping. These studies, however, utilize prediction of offline information, which may not always be available. Recently, [24], [31], [50], [52] leverage Lyapunov technique for data centers, but they do not consider water footprints. Furthermore, [24], [52] do not address service latency costs and their analysis primarily builds upon i.i.d./Markovian system dynamics (e.g., workload arrivals) which may not hold in practice.

More recently, [43] explores resource management approaches for optimizing real-time WUE by greedily exploring the spatio-temporal diversity of water efficiency, and hence the operational cost may be significantly increased (by 30%). For a single data center, [25] focuses on delay-tolerant batch jobs for reducing water consumption without considering spatial diversity of water efficiency, and hence it is not applicable for our study.

To sum up, our work takes an early step to address data centers' long-term water conservation, which is emerging as a critical concern amid extended droughts. WATCH can slash water footprints while only slightly increasing the operational cost, even compared to cost-driven COST.

7 Conclusion

In this paper, we took an early step towards long-term water conservation in data centers and proposed WATCH, a new water-aware workload management algorithm that can dynamically dispatch workloads to distributed data centers for capping water footprint. It was proved that using only online information, WATCH achieves a close-to-minimum operational cost compared to the optimal offline algorithm with future information, while bounding the potential violation of water capping. We also performed a trace-based simulation study to complement the analysis. The result was consistent with our analysis: it showed that WATCH significantly reduces the water consumption while only incurring a small operational cost increase. We also extended WATCH to cap carbon footprint as well as onsite water consumption for each data center.

References

[1] California ISO, http://www.caiso.com/.

[2] California water action plan, 2014, http://resources.ca.gov/california_water_action_plan/docs/Final_California_Water_Action_Plan.pdf.

[3] California drought, http://ca.gov/drought/.

[4] Drought monitor, http://droughtmonitor.unl.edu/.

[5] eBay, http://dse.ebay.com.

[6] Federal water efficiency requirements, http://www1.eere.energy.gov/femp/program/waterefficiency_requirements.html.

[7] AT&T Sustainability, http://www.att.com/gen/landing-pages?pid=24188.

[8] Data centers move to cut water waste, http://www.datacenterknowledge.com/.

[9] Data Center Knowledge: The Facebook Data Center FAQ, http://www.datacenterknowledge.com/the-facebook-data-center-faq/.

[10] Facebook data center dashboard, http://www.fbpuewue.com.

[11] Historical Weather, http://www.wunderground.com/.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[13] N. Deng, C. Stewart, D. Gmach, M. Arlitt, and J. Kelley. Adaptive green hosting. In *ICAC*, 2012.

[14] X. Dong, T. El-Gorashi, and J. M. Elmirghani. Green ip over wdm networks with data centers. *Journal of Lightwave Technology*, 29(12):1861–1880, 2011.

[15] J. Doyle, R. Shorten, and D. O'Mahony. Stratus: Load balancing the cloud for carbon emissions control. *IEEE Transactions on Cloud Computing*, 1(1):1–1, Jan 2013.

[16] Environmental Defense Fund. How cooling towers work, http://business.edf.org/projects/attwater/how-cooling-towers-work.

[17] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *ISCA*.

[18] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms. In *SIGMETRICS*, 2009.

[19] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M. A. Kozuch. Autoscale: Dynamic, robust capacity management for multi-tier data centers. *ACM Trans. Comput. Syst.*, 30(4):14:1–14:26, Nov. 2012.

[20] 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.

[21] M. Gattulli, M. Tornatore, R. Fiandra, and A. Pattavina. Low-emissions routing for cloud computing in ip-over-wdm networks with data centers. *IEEE Journal on Selected Areas in Communications*, 32(1):28–38, January 2014.

[22] Google's Data Center Efficiency. http://www.google.com/about/datacenters/.

[23] B. Guenter, N. Jain, and C. Williams. Managing cost, performance and reliability tradeoffs for energy-aware server provisioning. In *IEEE Infocom*, 2011.

[24] Y. Guo, Z. Ding, Y. Fang, and D. Wu. Cutting down electricity cost in internet data centers by using energy storage. In *Globecom*, 2011.

[25] M. A. Islam, K. Ahmed, , S. Ren, and G. Quan. Exploiting temporal diversity of water efficiency to make data center less 'thirsty'. In *ICAC*, 2014.

[26] KSL. New Utah NSA center requires 1.7M gallons of water daily to operate, http://www.ksl.com/?sid=25978926.

[27] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir, and M. Martonosi. Capping the brown energy consumption of internet services at low cost. In *IGCC*, 2010.

[28] M. Lin, Z. Liu, A. Wierman, and L. L. H. Andrew. Online algorithms for geographical load balancing. In *IGCC*, 2012.

[29] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska. Dynamic right-sizing for power-proportional data centers. In *IEEE Infocom*, 2011.

[30] Z. Liu, M. Lin, A. Wierman, S. H. Low, and L. L. Andrew. Greening geographical load balancing. In *SIGMETRICS*, 2011.

[31] H. Mahmud and S. Ren. Online resource management for data center with energy capping. In *Feedback Computing*, 2013.

[32] U.S. Dept. of Energy. Energy demands on water resources. Dec. 2006.

[33] Z. M. Mao, C. D. Cranor, F. Douglis, M. Rabinovich, O. Spatscheck, and J. Wang. A precise and efficient evaluation of the proximity between web clients and their local DNS servers. In *USENIX ATC*, 2002.

[34] T. Mastelic, A. Oleksiak, H. Claussen, I. Brandic, J.-M. Pierson, and A. V. Vasilakos. Cloud computing: Survey on energy efficiency. *ACM Comput. Surv.*, 47(2):33:1–33:36, Dec. 2014.

[35] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling "cool": Temperature-aware workload placement in data centers. In *USENIX ATC*, 2005.

[36] National Geographic. Water stress threatens future energy production, http://newswatch.nationalgeographic.com/2013/07/18/water-stress-threatens-future-energy-production/.

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

[38] Office of Governor in California. Governor brown declares drought state of emergency, 2014, http://gov.ca.gov/news.php?id=18368.

[39] M. Polverini, A. Cianfrani, S. Ren, and A. Vasilakos. Thermal-aware scheduling of batch jobs in geographically distributed data centers. *IEEE Transactions on Cloud Computing*, 2(1):71–84, Jan 2014.

[40] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *SIGCOMM*, 2009.

[41] 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.

[42] C. Ren, D. Wang, B. Urgaonkar, and A. Sivasubramaniam. Carbon-aware energy capacity planning for datacenters. In *MASCOTS*, 2012.

[43] S. Ren. Optimizing water efficiency in distributed data centers. In *Cloud and Green Computing*, 2013.

[44] SPX Cooling. Water usage calculator, http://spxcooling.com/green/leed/water-usage-calculator/.

[45] The Green Grid. Water usage effectiveness (WUE): A green grid data center sustainability metric. *Whitepaper*, 2011.

[46] Uptime Institute. Data center industry survey, 2013, http://uptimeinstitute.com/2013-survey-results.

[47] U.S. EPA. http://www.epa.gov/cleanenergy/documents/egridzips/eGRID_9th_edition_V1-0_year_2010_Data.xls.

[48] U.S. Green Building Council. Leadership in energy & environmental design, http://www.usgbc.org/leed.

[49] V. Vasudevan, A. Phanishayee, H. Shah, E. Krevat, D. G. Andersen, G. R. Ganger, G. A. Gibson, and B. Mueller. Safe and effective fine-grained tcp retransmissions for datacenter communication. *SIGCOMM Comput. Commun. Rev.*, 39(4):303–314, Aug. 2009.

[50] 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.

[51] Q. Zhang, M. Zhani, R. Boutaba, and J. Hellerstein. Dynamic heterogeneity-aware resource provisioning in the cloud. *IEEE Transactions on Cloud Computing*, 2(1):14–28, Jan 2014.

[52] Z. Zhi, F. Liu, Y. Xu, R. Zou, H. Xu, J. C. S. Lui, and H. Jin. Carbon-aware load balancing for geo-distributed cloud services. In *MASCOTS*, 2013.

[53] X. Zhu, L. Yang, H. Chen, J. Wang, S. Yin, and X. Liu. Real-time tasks oriented energy-aware scheduling in virtualized clouds. *IEEE Transactions on Cloud Computing*, 2(2):168–180, April 2014.

8 Appendix: Terms in Water Management

Wet-bulb temperature: It is the temperature of air with 100% water saturation. It indicates the lowest temperature achievable by water evaporation. For cooling towers, it is the theoretical lowest temperature to which the water can be cooled down.

Waterside economizer: For the cooling systems that use chilled water to transfer heat from server room to the outside, water entering the server room is called "chilled water" (e.g., cooled down by the mechanical chiller) and water leaving the server room becomes "warm". Waterside economizer refers to using cooling tower, instead of mechanical chillers, to cool down the warm water returned from the data center server room. Naturally, waterside economizer is only applicable when the outside temperature is cold enough, and thus it is mostly used in winter.

Water blown-down: The cooling tower losses water through evaporation to dissipate heat to the environment. In the process, the mineral concentration increases as more and

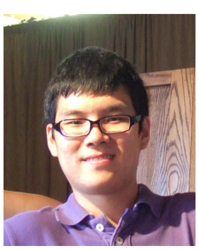
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2015.2453982, IEEE Transactions on Cloud Computing

more water is evaporated. To avoid mineral's accumulation in the water circulation system, concentrated cooling tower water is drained at regular intervals, which is called water blown-down. This is one of the two major components of data center direct water consumption (the other one is evaporation).

Cycle of concentration: To avoid mineral's building up, the water cycle in a cooling tower required to be blown down at regular intervals. Naturally, if the original water source has more minerals, we can circulate the water fewer times before the mineral concentration becomes harmful for the plant. Cycle of concentration refers to the maximum number of cycles the water can be used before it is blown down. As blown-down is one of the principal components of water consumption, more cycles of concentration can lead to lower water consumption.

Water withdrawal v.s. water consumption: Water withdrawal refers to the water that is withdrawn but later may return to the same source. For example, in once-through cooling systems (used for thermal power plants), water is usually withdrawn from a nearby lake or river, but the water simply flows through the cooling system and then most of the water still returns to the source (with a small fraction of water evaporated to dissipate heat). Water consumption, on the other hand, refers to the water that is actually "lost" and not returned to the source. For example, water evaporation and water blown-down (to, e.g., sewage systems) are considered "consumption", since the water is not returned directly to its source.

Mohammad A. Islam is doing his Ph.D. in Electrical Engineering at Florida International University, Miami, Florida. He received his B.Sc. in Electrical and Electronics Engineering from Bangladesh University of Engineering and Technology in January 2008. He is a member of the Sustainable Computing Group (SCG) at FIU led by Dr. Shaolei Ren. His research interests include data center resource management, cloud computing, and sustainability for IT.



Shaolei Ren received his B.E., M.Phil. and Ph.D. degrees, all in electrical engineering, from Tsinghua University in July 2006, Hong Kong University of Science and Technology in August 2008, and University of California, Los Angeles, in June 2012, respectively. Since August 2012, he has been with School of Computing and Information Sciences, Florida International University, as an Assistant Professor. His research interests include cloud computing, data center resource management, and sustainable computing. He received the Best Paper Award from IEEE International Conference on Communications in 2009 and from International Workshop on Feedback Computing in 2013.

Gang Quan (M02-SM'10) received the B.S. degree from the Tsinghua University, Beijing, China, the M.S. degree from the Chinese Academy of Sciences, Beijing, and the Ph.D. degree from the University of Notre Dame, Notre Dame, IN. He is currently an Associate Professor with the Electrical and Computer Engineering Department, Florida International University, Miami. His research interests includes real-time system, power/thermal aware design, embedded system design, advanced computer architecture and reconfigurable computing. Dr. Quan is the recipient of a National Science Foundation Faculty Career Award. He also won the Best Paper Award from the 38th Design Automation Conference. His paper was also selected as one of the Most Influential Papers of 10 Years Design, Automation, and Test in Europe Conference (DATE) in 2007. Dr. Quan is a senior member of IEEE.

Muhammad Zeeshan Shakir has been an assistant research scientist at Texas A&M University at Qatar in Doha, Qatar, since July 2012. He received the Ph.D. in electronic and electrical engineering from the University of Strathclyde, Glasgow, UK, in 2010. From January 2006 to September 2009, he was the joint recipient of an industrial research fund and a prestigious overseas research scholarship by the University of Strathclyde. His research interests include design and deployment of diverse Green wireless communication systems. He has published more than 75 technical journal and conference papers and has contributed to six books, all in reputable venues. He is co-author of two research monographs. Most of his research has been sponsored by Qatar National Research Fund and national industrial partners. He has been serving as a lead guest editor for IEEE Communication Magazine and IEEE Wireless Communications Magazine. He has been serving as co-chair of several special sessions/workshops and symposia in flagship conferences, such as ICC and GlobalSIP.

Athanasios V. Vasilakos (M?00?SM?11) is currently a Professor at Lulea University of Technology, Sweden. He served or is serving as an Editor for many technical journals, such as the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT; IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY; IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS?PART B:CYBERNETICS; IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE; IEEE TRANSACTIONS ON CLOUD COMPUTING, ACM Transactions on Autonomous and Adaptive Systems; the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS(May 2009,Jan 2011,March 2011 issues). He is also General Chair of the European Alliances for Innovation(www.eai.eu).