CoolAir: Temperature- and Variation-Aware Management for Free-Cooled Datacenters

Íñigo Goiri[†*], Thu D. Nguyen[‡], and Ricardo Bianchini[†‡]

[‡]Rutgers University
{tdnguyen,ricardob}@cs.rutgers.edu

[†]Microsoft Research
{inigog,ricardob}@microsoft.com

Abstract

Despite its benefits, free cooling may expose servers to high absolute temperatures, wide temperature variations, and high humidity when datacenters are sited at certain locations. Prior research (in non-free-cooled datacenters) has shown that high temperatures and/or wide temporal temperature variations can harm hardware reliability.

In this paper, we identify the runtime management strategies required to limit absolute temperatures, temperature variations, humidity, and cooling energy in free-cooled datacenters. As the basis for our study, we propose CoolAir, a system that embodies these strategies. Using CoolAir and a real free-cooled datacenter prototype, we show that effective management requires cooling infrastructures that can act smoothly. In addition, we show that CoolAir can tightly manage temperature and significantly reduce temperature variation, often at a lower cooling cost than existing free-cooled datacenters. Perhaps most importantly, based on our results, we derive several principles and lessons that should guide the design of management systems for free-cooled datacenters of any size.

Categories and Subject Descriptors C.m [*Computer Systems Organizations*]: Miscellaneous; D.4.m [*Operating systems*]: Miscellaneous

Keywords Thermal management, energy management, free cooling, datacenters.

* This work was done while Íñigo was at Rutgers University.

ASPLOS '15, March 14–18, 2015, Istanbul, Turkey..

ACM 978-1-4503-2835-7/15/03…$15.00.
http://dx.doi.org/10.1145/2694344.2694378

1. Introduction

Recently, datacenter operators have started building their datacenters in cold and dry climates, where outside air can be used to cool the servers [14, 18]. In this approach, known as "air-side economizer" or simply "free" cooling, fans blow the cool outside air into the datacenter so that it reaches the server inlets, while the warm air heated up by the servers is guided back outside [33]. Free cooling significantly reduces energy consumption, and may obviate the need for water chillers, cooling towers, and air conditioners (ACs) in many geographical locations. Free cooling can be useful *across the spectrum of datacenter sizes,* from small enterprise datacenters to warehouse-scale datacenters run by large Internet companies.

Motivation. In recent years, free cooling has received attention in the popular media [28, 29] and in the scientific literature [2, 12, 15, 22, 23, 27, 41]. There is a consensus that free cooling reduces cooling costs and should be used whenever and wherever possible. However, the potential impact of free cooling on hardware reliability is not yet fully understood. For example, researchers have recently studied the impact of temperature on hardware (especially, hard disk drive) reliability in *non*-free-cooled datacenters [10, 34, 36] with conflicting results. Pinheiro *et al.* [34] studied Google datacenters and observed the impact of absolute disk temperature on reliability to be small up to roughly 50°C. El-Sayed *et al.* [10] studied a variety of datacenters (including some from Google) and reached a similar conclusion, but found wide temporal temperature variations to increase sector errors more significantly and consistently. Sankar *et al.* [36] studied Microsoft datacenters and observed that absolute disk temperature has a significant impact on reliability, whereas wide temperature variation does not.

Regardless of how these conflicting results will be reconciled, it is clear that free-cooled datacenters may expose servers to high absolute temperatures and/or wide temporal temperature variations at locations where outside temperatures are high and/or highly variable. At these locations, free-cooled datacenters would exhibit higher temperatures and wider temperature variations than those shown in the disk reliability papers we cite above. However, for latency reasons or other restrictions on siting (*e.g.*, an enterprise's desire to build its datacenter next to its offices), it may be desirable to build free-cooled datacenters at such locations.

To the best of our knowledge, there has been no *publicly available* study of the temperatures and variations to which servers are exposed in free-cooled datacenters. Similarly, there has been no study of techniques that can manage absolute temperature and temperature variations in those datacenters. Importantly, such techniques should keep the cooling energy consumption low, as many locations exhibit a tradeoff between the cooling energy savings due to free cooling and hardware maintenance and replacement costs.

This paper fills these large gaps in the literature. Specifically, we propose CoolAir, a workload and cooling management system for free-cooled datacenters. As a case study, we apply CoolAir to our real free-cooled datacenter prototype, called Parasol [17]. Parasol combines free cooling with a direct-expansion (DX) AC, and embodies a commercial feedback-driven cooling controller that periodically selects the cooling regime (*e.g.*, free cooling at 50% fan speed, or free cooling off with AC on). Despite this controller, Parasol exhibits high inlet air temperatures and wide daily inlet air temperature variations. We refer to Parasol running this controller as the "baseline" system.

CoolAir. CoolAir manages absolute temperature, temperature variation, relative humidity, and cooling energy in free-cooled datacenters. It uses machine learning, simulation, and weather forecasts to model and predict temperatures, humidity, and cooling energy. To achieve its goals, CoolAir manages the workload placement and schedule, the servers' power state, as well as the cooling regime. In contrast, the baseline system only controls the cooling regime and does so in a more limited fashion (*e.g.*, without concern for future outside temperatures or internal temperature variations).

Although we study CoolAir for Parasol and Hadoop workloads, it can be easily adapted to other cooling infrastructures and software systems. Interestingly, our multiple day-long real experiments with CoolAir on Parasol show that it is impossible to control temperature variation with Parasol's cooling infrastructure; the cooling units react too abruptly during certain regime transitions.

Evaluation. To evaluate CoolAir for an entire year, we build a simulator of Parasol and *validate* it against multiple (day-long) real experiments on Parasol. To study a more controllable setup, we simulate a free cooling unit with fine-grained fan speed ramp up, and a variable-speed AC. These types of cooling units are available commercially [7, 11, 37]. Using the simulator, we study CoolAir in detail for five locations: Iceland (cold year-round), Chad (hot year-round), Santiago de Chile (mild year-round), Singapore (hot and humid year-round), and Newark (hot in the summer, cold in the winter). Like Newark, Parasol is located in New Jersey, USA. We also explore 1500+ locations world-wide.

Our results for the manageable version of Parasol demonstrate that CoolAir can limit temperatures and significantly reduce daily variations, while keeping Power Usage Efficiencies (PUEs) low, compared to the baseline system. Interestingly, managing temperature variation incurs a substantial cooling energy penalty. Nevertheless, CoolAir's management of cooling energy produces cooling costs that are often even lower than those of the baseline system. In fact, CoolAir is broadly useful: at locations where it reduces variations significantly, it does so at little or no cost in PUE; at locations where it tends to reduce variations less substantially, it lowers PUEs. Importantly, our results quantify the energy cost of lowering absolute temperature and temperature variation, as a function of location. Overall, our results demonstrate that *CoolAir can effectively manage temperatures (and consequently disk reliability) and cooling energy all over the world.*

Principles and lessons. From our case study, we derive *general* lessons that should apply to free-cooled datacenters of any size. For example, we find that effective management requires fine-grain management knobs, and temperature setpoints based on outside temperature. Moreover, we find that *existing energy-driven spatial and temporal workload scheduling techniques (*e.g.*, [2, 22, 27, 30, 32]) have a negative impact on variation,* suggesting that these techniques should be avoided in free-cooled datacenters.

Contributions. In summary, our main contributions are:

1. We show that free cooling may expose servers to high absolute temperatures and wide daily temperature variations.

2. We propose CoolAir, a general workload and cooling manager for free-cooled datacenters. No prior work has considered temperature variation or future outside temperatures in managing internal temperatures. Moreover, some existing thermal management techniques actually *increase* variation.

3. As a case study, we evaluate CoolAir using day-long experiments on a real free-cooled datacenter prototype, and on a simulated system with a more sophisticated cooling infrastructure for a year at more than 1500 locations.

4. We derive principles and lessons for the management systems of free-cooled datacenters of any size.

2. Background

Datacenters have traditionally used different cooling infrastructures depending on their sizes. Large datacenters have often used water chillers and cooling towers outside the datacenter, and air handlers inside of it. Medium datacenters often forgo the cooling tower, but retain the chiller and air handling units. Small datacenters often replace the chiller and air handler with in-datacenter DX ACs. These ACs utilize compressors to remove heat. All datacenters now create "cold aisles" in front of the servers, and isolate them from the "hot aisles" behind the servers.

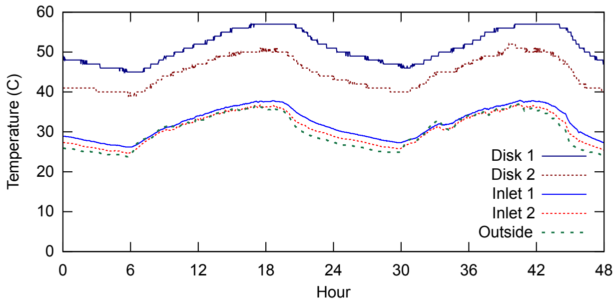

Figure 1. Disk, inlet, and outside temps under free cooling.

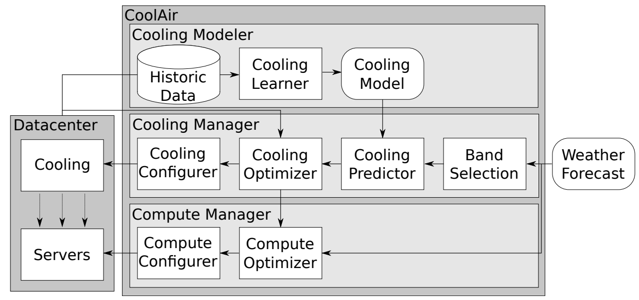

Figure 2. Architecture. Rectangles with square corners represent modules, whereas round corners represent derived or input data.

Recently, datacenters of all sizes have started to leverage free cooling by using fans to blow cool (filtered) outside air into the datacenter. Again using fans, the warm return air is guided back out of the datacenter. When outside temperature and humidity are appropriate, free cooling obviates the need to run power-hungry water chillers or compressors. In fact, in cool climates, free cooling enables the forgoing of chillers and compressors altogether. In warmer climates, some free-cooled datacenters also apply adiabatic cooling (via water evaporation, within the humidity constraint) to lower the temperature of the outside air before letting it reach the servers. Another approach (the one we study in this paper) is to combine free cooling with more power-hungry, "backup" cooling (*e.g.*, ACs), for use when outside air temperature or humidity is too high.

Free cooling operation. Typically, the cooling system uses a (feedback-based) controller that modulates the fan speed to keep the inside temperature below a setpoint. The controller is also responsible for (turning free cooling off and) activating the backup cooling system when necessary.

3. CoolAir

The main goal of CoolAir is to manage absolute temperatures and daily temperature variations in free-cooled datacenters, while keeping energy consumption low. In addition, CoolAir manages relative humidity and the rate of air temperature change per hour. CoolAir achieves its goals by intelligently (1) selecting the best setpoint for the inlet air temperatures on a daily basis (leverages predictions of outside temperatures); (2) placing the offered workload around the datacenter (spatial placement); (3) possibly scheduling the workload for later execution (temporal scheduling, when the workload is deferrable); and (4) periodically selecting the best cooling regime according to all air temperature and humidity sensors. It also decides how many servers to keep active. We refer to datacenter "utilization" as the fraction of active servers; *e.g.*, 50% utilization means that half of the servers are active and the other half is asleep (in ACPI's S3 state).

CoolAir assumes: at least one air temperature sensor and at least one humidity sensor outside the datacenter; the datacenter is organized into "pods" of servers (*i.e.*, sets of servers that are spatially close) [30]; at least one air temperature sensor representing the inlet temperature of each pod; and a humidity sensor in one of the cold aisles.

Importantly, CoolAir manages air temperatures and humidity to prevent disk reliability from suffering under free cooling. Disks are the most sensitive components to temperature and temperature variation. Figure 1 illustrates the relationship between outside air temperature, inside inlet air temperature, and disk temperature during Parasol's free cooling operation. The figure plots the lowest and highest disk temperatures on July 6th and 7th 2013, when we ran a workload that constantly left the disk 50% utilized. Clearly, there is a strong correlation between air and disk temperatures. *Any datacenter exposed to outside temperatures would exhibit this same correlation.*

Figure 2 overviews the CoolAir architecture. It comprises three main components: the *Cooling Modeler*, the *Cooling Manager*, and the *Compute Manager*. The following subsections describe each component in detail.

3.1 Cooling Modeler

The Cooling Modeler is responsible for modeling temperatures and the cooling system. It collects air temperature and humidity for each available sensor, the utilization of each server, the current status of the cooling infrastructure, and the cooling power currently being consumed. The *Cooling Learner* then executes a linear regression algorithm to learn a model (the *Cooling Model*) for the thermal behavior in multiple locations of the cold aisles and the humidity in one location of the cold aisles, as a function of many parameters. The Cooling Modeler runs offline and only once, after enough data has been collected under the default cooling controller.

Temperature. We model the thermal behavior of each area of interest inside a datacenter, *i.e.* the location of a temperature sensor, using a set of equations of the form $T = F(I)$, where the predicted temperature T is a linear function of the inputs I. The inputs are the current and last inside air temperature (at the sensor's location), the current and last outside air temperature, the current and last fan speed of the free cooling system, the current datacenter utilization, the product of the current fan speed and the current inside air temperature, and the product of the current fan speed and the current outside air temperature. Note that we use composed inputs (*e.g.*, current fan speed $\times$ current temperature) to allow linear regression.

Using machine learning and extensive monitoring data, CoolAir generates *a distinct function F for each possible cooling regime and transition between regimes*, including neither free cooling nor air conditioning, free cooling at different speeds, AC compressor-on, and transition from free cooling to AC compressor-on. For simplicity, we assume that a given datacenter utilization always causes the same servers to be active.

Humidity. We similarly model absolute humidity inside the datacenter using a set of equations $H = G(I')$, where the humidity H is a linear function of the inputs I'. The inputs are the current inside air humidity, the current outside air humidity, the current fan speed of the free cooling system, the product of the fan speed and the inside humidity, and the product of the fan speed and the outside humidity. Again, *CoolAir produces a function G for each cooling regime and transition between regimes.* It uses the predicted inside air temperature (at a chosen location) to convert the predicted absolute inside air humidity to a relative inside air humidity.

Cooling power. We model it as a constant amount drawn in each regime: (1) per each fan speed during free cooling operation; and (2) when free cooling is off and backup cooling is on (possibly in different operating modes).

3.2 Cooling Manager

The Cooling Manager selects the best operating region, selects and installs good cooling configurations, and produces cooling predictions for the Compute Manager. Since CoolAir seeks to limit variation, it uses a target *range (or band) of temperatures* within which it tries to keep inlet temperatures.

Temperature band selection. The Cooling Manager first selects the band that would be easiest to maintain throughout the day while limiting temporal variation. For example, on a cool day, it can maintain a relatively low temperature range by alternating between free cooling and simply closing the datacenter (*i.e.*, using neither free cooling nor backup cooling). On a warm day, it can maintain a relatively higher temperature by using free cooling, and backup cooling only if necessary.

CoolAir selects the band by querying a Web-based weather forecast service to find the hourly outside temperature predictions at the datacenter's location for the rest of the day. With these predictions, it selects a band of temperatures *Width* degrees wide around the average predicted outside temperature for the day plus an *Offset*. Intuitively, the width of the band should be selected based on the expected temperature variations over any one day, and how much energy one wants to spend limiting temperature variations. Since the outside air naturally heats up before arriving at the server inlets, the *Offset* corresponds to the typical difference between outside and inside air temperatures (*e.g.*, see Figure 1, where *Offset*

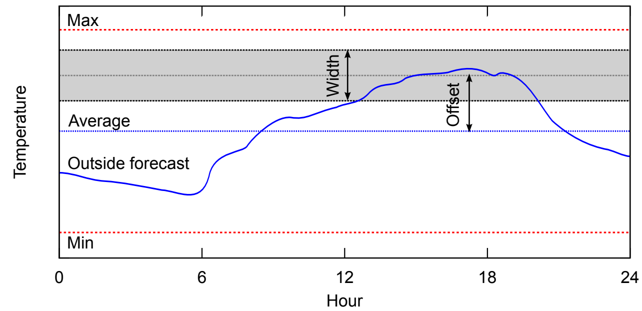

Figure 3. Temperature band selection in CoolAir.

is 2.5°C). Figure 3 illustrates the band selection. However, no part of the band can be higher than the *Max* temperature or lower than the *Min* temperature; the band slides back just below *Max* or just above *Min* in those cases.

Selecting a temperature band for each day independently of other days could contribute to increases in long-term temperature variation. However, this is not a problem since: (1) variation is most harmful to reliability when it happens quickly (*e.g.*, ASHRAE recommends a temperature change rate of at most 20°C/hour [3]); and (2) we set *Width* wide enough that transitions between days are smooth, *i.e.* bands for consecutive days almost always overlap.

Cooling Optimizer. This module periodically (every 10 minutes) selects the best cooling regime to use for the next period. It uses the *Cooling Predictor* (described below) and information about the current datacenter utilization, air temperatures, and humidity to predict the thermal behavior that would result from using each of the regimes for the next period (10 minutes). To select one of these choices, the Cooling Optimizer uses a simple *utility function*.

The utility function considers the predictions for: (1) absolute temperature; (2) temperature variation; (3) relative humidity; and (4) cooling energy. To guide the system towards the CoolAir goals, we assign a penalty to any violation of these goals. Specifically, the following violations all carry the same penalty: each 0.5°C higher than the maximum temperature threshold, each 1°C of temperature variation higher than 20°C/hour, each 0.5°C outside of the temperature band, each 5% of relative humidity outside of the humidity band, and turning on the AC at full speed. The overall function value for each cooling regime is the sum of the penalties for the sensors of all active pods. In Section 5, we also consider simplified versions of CoolAir that disregard certain components of the utility function.

Cooling Predictor. The Cooling Optimizer calls the Cooling Predictor when it needs temperature and relative humidity predictions for a cooling regime it is considering. The Predictor then uses the Cooling Model to produce the predictions. However, as the Cooling Model predicts temperatures for a short term, the Cooling Predictor has to use it repeatedly (each time passing the results of the previous use as input).

Cooling Configurer. This is the only module that interacts directly with the cooling infrastructure, according to the behavior the Cooling Optimizer wants to achieve.

3.3 Compute Manager

The Compute Manager is responsible for the execution of the workload, and for activating and deactivating servers.

Compute Optimizer. This module optimizes the set of active servers and the *spatial placement* of the workload. To do so, it uses a ranking of pods in terms of their potential for heat recirculation.[1] This ranking comes from the Cooling Modeler, which creates it by observing changes in inlet temperature when load is scheduled on each pod.

CoolAir selects the set of servers that are most prone to heat recirculation as targets for the current workload. Although this may seem counter-intuitive, this approach makes it easier to manage temperature variation. The reason is that lower recirculation pods tend to be more exposed to the effect of the cooling infrastructure and, thus, may experience wider variations. In Section 5, we consider simplified versions of CoolAir that make different activation/placement decisions.

For deferrable workloads, CoolAir also performs *temporal scheduling*, *i.e.* it schedules the jobs that have already arrived 24 hours into the future. This scheduling may determine that certain jobs should be delayed. However, CoolAir will not delay any job beyond its user-provided *start deadline*. Within these deadlines, CoolAir tries to place as much load as possible during periods when the hourly predictions of outside air temperature for the day are within its temperature band. It does not temporally schedule jobs for a day, if (1) the band needs to slide back below *Max* or above *Min*, or (2) the band does not overlap with predicted outside temperatures (outside temperatures will constantly be higher than *Max* or lower than *Min*). Temporal scheduling provides no benefits for such days.

Compute Configurer. This module configures the number of active servers and is the only one that interacts with the system running on the datacenter (Hadoop in this paper).

4. CoolAir for Parasol

CoolAir can be adapted to any free-cooled datacenter. In this section, we describe its implementation for Parasol.

4.1 Parasol and Its Cooling Infrastructure

We built Parasol for research on several topics, including free cooling and wimpy servers. Parasol combines free cooling with DX air conditioning, allowing it to be used in a wide range of climates, as we study in this paper.

Internal view. Parasol comprises a small custom container (7'×12') that houses the servers. Figure 4 shows the internal

[1] Heat recirculation is a feature (not a bug) in free-cooled datacenters, and is used to increase temperature (when the outside air is too cold) and decrease humidity (when the outside air is too humid).

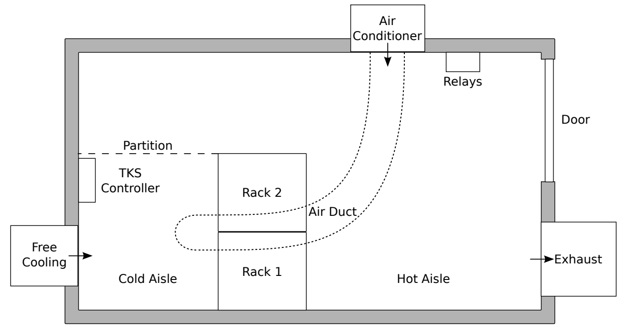

Figure 4. Internal layout of Parasol.

layout of the container in scale. The free cooling unit (Dantherm Flexibox 450) and its controller (TKS 3000) appear on the left, the servers in the two racks in the middle, the exhaust damper on the right, and the AC (Dantherm iA/C 19000) at the top-right part of the figure. Using simple partitions (the dashed line), we create a small, sealed cold aisle in front of the servers, and a "wind-tunnel" effect between the free cooling unit, the servers, and the damper. To seal the cold aisle, we also place a partition at the top of the racks (not shown). The location of the AC is not ideal, requiring an air duct (the snake-like shape across the figure) into the cold aisle. The AC's hot air intake is above the end of the duct in the hot aisle. The sealed cold aisle minimizes hot air recirculation during free cooling or AC operation. When the container is closed and both cooling units are off, recirculation occurs around the partitions and through any sleeping servers.

System operation. The TKS controller manages Parasol's cooling. It selects the cooling mode based on how the outside temperature relates to a configurable setpoint SP (25°C by default). Below the setpoint, it operates in the Low Outside Temperature (LOT) mode, and uses free cooling as much as possible. Above the setpoint, the TKS starts operating in the High Outside Temperature (HOT) mode. It then closes the damper, turns the free cooling unit off, and turns the AC on. The TKS uses hysteresis (1°C) around the setpoint. The AC operates in cycles: it stops the compressor when the inside temperature is lower than $SP - 2$°C, and starts it when the inside temperature is higher than SP. The AC consumes either 135W (fan only) or 2.2kW (compressor and fan on).

In LOT mode, the TKS operates based on the temperature of a control sensor located in a typically warmer area in the cold aisle. When the temperature at this sensor is low, the TKS turns free cooling off and closes the container, which increases temperatures via recirculation. When the internal temperature is between SP and $SP - P$ (P is set to 5°C by default), the TKS uses free cooling, and selects the fan speed based on the difference between the outside and the inside temperatures. The closer the two temperatures are, the faster the fan blows. The minimum speed is 15% of the maximum speed. If the outside temperature is much lower than the inside temperature, the fan blows slowly but the inside temperature still drops fast due to the cold outside air. The free cooling unit draws between 8W and 425W, depending on fan speed.

The above description identifies the main cooling regimes in Parasol: (1) free cooling with a fan speed above 15%; (2) air conditioning with the compressor on or off; or (3) neither (the datacenter is closed).

Though a successful commercial product, the TKS lacks key CoolAir features: daily temperature setpoint selection based on future outside temperature; all-sensor temperature, variation, and humidity control; spatial and temporal workload management; and energy management. In the absence of these features, Parasol's disk reliability could suffer severely. Our evaluation compares CoolAir to a baseline that extends the TKS control scheme, as we describe in Section 5.1.

4.2 CoolAir Implementation

Monitoring Parasol. Parasol has one air temperature sensor for each server pod, which includes the servers that behave similarly (*e.g.*, same temperature changes, same potential for recirculation) in response to the cooling regimes. Parasol also monitors humidity with one sensor in each aisle. It uses the periodic temperature and humidity measurements in modeling the thermal behavior (Section 3).

Cooling Configurer for Parasol. CoolAir translates its desired actions into changes to the TKS temperature setpoint SP. To translate the temperature band, it sets the top of the band to be SP and *Width* to be the P value. By changing the TKS setpoint, we can also turn off the free cooling (which stops the flow of air into and out of Parasol), change the free cooling fan speed, and activate the AC (turning free cooling off). When the AC is on, we can also control its internal setpoint, and whether the compressor should be off.

Compute Configurer for Hadoop. In our implementation, CoolAir manages a slightly modified version of Hadoop [17]. The modification enables energy management by implementing three server power states: active, decommissioned, and sleep. Active and sleep (the S3 state) are self-explanatory. The decommissioned state is an intermediate state that prevents new jobs from starting on the server. We configure our Hadoop setup to the Covering Subset scheme [24], *i.e.* we store a full copy of the dataset on the smallest possible number of servers; any server out of the Covering Subset can be sent to sleep without affecting data availability.

The Compute Configurer transitions servers between the server power states in three ways: (1) it transitions any active server that need not be active but still stores (temporary) data required by running jobs to the decommissioned state. In a following iteration, if the data stored at the decommissioned server is no longer needed, the Configurer transitions it to the sleep state; (2) it transitions any active server that need not be active and does not store relevant data to the sleep state; and (3) it transitions sleeping servers to the active state if they are required for computation during an iteration. The Configurer keeps the Covering Subset active at all times.

In the absolute worst case, by transitioning power states, the Compute Configurer could cause a set of disks to power-cycle every 20 minutes, *i.e.* 3 times per hour. Fortunately, modern disks use load/unload technology [21], which enables them to power-cycle at least 300,000 times without failure. This means that disks can be power-cycled 8.5 times per hour on average, during their 4-year typical lifetime. For our workloads, no disk gets power-cycled more than 2.2 times per hour on average.

Data collection and model learning. To create our models, *we collected temperature, humidity, power consumption data from Parasol for 1.5 months.* To get a richer dataset within this period of time, we intentionally generated extreme situations by changing the cooling setup (*e.g.*, temperature setpoint), and monitored the resulting behaviors. We then use regression on this dataset to generate the specific set of functions F and G that together comprise the models (one model per cooling regime or transition between regimes) for predicting air temperature and humidity (Section 3.1). We use a similar approach to build the power model. We use Weka [19] to generate these regressions. For behaviors that are non-linear (*e.g.*, power consumption as a function of free cooling speed), we generate piece-wise linear models using M5P. For linear behaviors, we try linear and least median square approaches and pick the one with the lowest error. Prior works [25, 31] have shown regressions to work well for this problem.

Validation of the temperature and humidity models. *We compare the predicted temperatures to measured values in Parasol, during two entire (and non-consecutive) days* that were not in the learning dataset. Figure 5 plots the CDFs for the prediction error (in °C) for four cases: (1) *2-minutes:* the absolute value of the difference between the predicted temperature 2 minutes into the future and the measured temperature; (2) *2-minutes no-transition:* same as (1) except that we consider only 2-minute intervals that did not involve a transition in cooling regime; (3) *10-minutes:* the absolute value of the difference between the predicted temperature 10 minutes into the future and the measured temperature; and (4) *10-minutes no-transition:* same as (3) except that we consider only 10-minute intervals that did not involve a regime transition.

These results show that the temperature models are quite accurate, especially for periods without transitions between cooling regimes. Specifically, without transitions, 95% of the 2-minutes and 90% of the 10-minutes predictions are within 1°C of measured values. Even when including transitions, over 90% of the 2-minutes and over 80% of the 10-minutes predictions are within 1°C of measured values.

We used the same approach to validate our relative humidity models. Again, the results show that the models are quite accurate: 97% of our predictions are within 5% (in absolute terms) of the measured humidities.

CoolAir version	Workload type	Utility function	Spatial placement	Temporal scheduling
Temperature	Non-deferrable	Lower max temp + energy + humidity	Low recirculation	No
Variation	Non-deferrable	Adaptive band (max 30°C) + humidity	High recirculation	No
Energy	Non-deferrable	Max temp (30°C) + energy + humidity	Low recirculation	No
All-ND	Non-deferrable	Adaptive band (max 30°C) + energy + humidity	High recirculation	No
All-DEF	Deferrable	Adaptive band (max 30°C) + energy + humidity	Low recirculation	Yes

Table 1. CoolAir versions. All-ND: CoolAir for non-deferrable workloads. All-DEF: CoolAir for deferrable workloads.

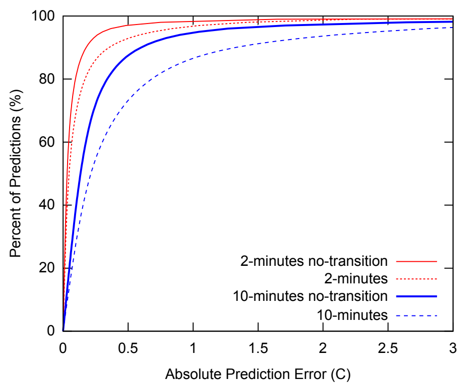

Figure 5. Modeling errors for 5/1/13 and 6/20/13.

4.3 Example Behavior

As an example of the behavior of CoolAir for Parasol, consider Figures 7(a) and 7(b). Figure 7(b) illustrates how CoolAir tried to keep the inlet temperatures between 24 and 29°C, given the workload described in Figure 7(a), on 06/15/2013. For example, around 7:30am, the inlet temperatures were starting to exceed 29°C and the outside temperature was low enough, so CoolAir opened up Parasol for free cooling (lightly shaded area labeled FC). CoolAir did not react to lower the temperatures sooner because this would have caused a higher utility function penalty. It then modulated the fan speed to regulate temperature until around 5:45pm, when the workload increased and inlet temperatures again started to exceed 29°C. At that point, it activated the AC (darkly shaded area labeled AC), since the workload was too heavy for it to lower temperatures enough with free cooling. As we show in Section 5, CoolAir can do an even better job of managing temperatures when the cooling infrastructure can react more smoothly to changes in cooling regime (Figure 7(d)).

5. Evaluation

5.1 Methodology

CoolAir and baseline system configurations. We configure CoolAir with *Offset* = 8°C, which is the offset that we normally observe in Parasol. We set *Width* = 5°C, as narrower bands tend to make it harder to control temperature variations (higher cooling energy and more regime changes) and wider bands needlessly allow temperatures to vary. We set *Min* and *Max* to 10°C and 30°C, respectively. We configure CoolAir to keep the relative humidity below 80% and air temperature changes below 20°C/hour. These settings roughly correspond to the "allowable" values suggested by ASHRAE [3].

We study multiple versions of CoolAir, called *Temperature*, *Variation*, *Energy*, *All-ND*, and *All-DEF*. The Temperature version only focuses on limiting absolute air temperatures below a low setpoint. This version represents what energy-aware thermal management systems do in non-free-cooled datacenters today. We set the setpoint to the lowest value that achieves the same PUE as the baseline system (described below). The Variation version focuses solely on limiting the amount of air temperature variation. The Energy version manages absolute temperatures, while attempting to conserve cooling energy. However, it does not manage temperature variation. This version is the same as Temperature, but it targets a higher maximum temperature to lower the cooling energy further. The All-ND version is the complete CoolAir implementation, which manages absolute temperatures, temperature variation, and cooling energy. All versions above are for non-deferrable workloads. We also study All-DEF, which manages absolute temperatures, temperature variation, and cooling energy for deferrable workloads. Table 1 lists the characteristics of each version.

We compare these versions to a baseline system that extends Parasol's default control scheme (Section 4.1) in two ways: (1) we set the setpoint to 30°C, instead of the default 25°C; and (2) we add humidity control to it, with a maximum limit of 80% relative humidity. These extensions make the baseline more efficient and comparable to CoolAir.

Parasol setup. Parasol currently hosts 64 half-U servers, each of which has a 2-core Atom D525MW CPU, 4GB of DRAM, a 250GB hard disk, and a 64GB solid-state drive. CoolAir's Compute Manager runs on one of these servers. Each server draws from 22W to 30W. Parasol also hosts a 4-core Xeon server, which runs Cooling Modeler and Cooling Manager. Parasol's sensors are accurate to within 0.5°C.

Workloads. We run our modified version of Hadoop on the 64 servers. We use a remote client to submit the Hadoop jobs. Each real experiment runs for a full day.

We study two widely different Hadoop traces, called "Facebook" and "Nutch". Facebook comes from a larger trace of 600 machines at Facebook [8]. We use Statistical Workload Injector for MapReduce (SWIM) [9] to generate a day-long, scaled-down version of the trace for 64 machines. In the resulting trace, each job comprises 2–1190 map tasks and 1–63 reduce tasks. There are roughly 5500 jobs and 68000 tasks.

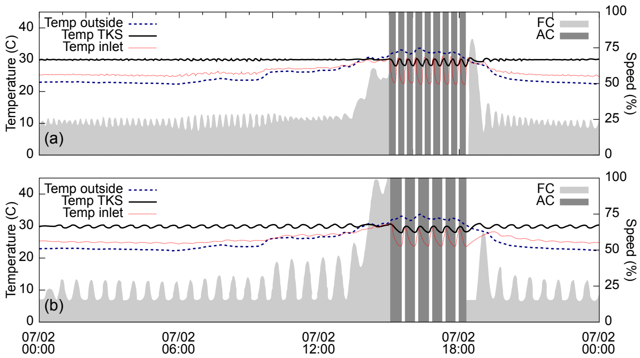

Figure 6. Real (a) and Real-Sim (b) baseline runs on 7/2/13.

The map phase of each job takes 25–13000 seconds, whereas the reduce phase takes 15–2600 seconds. Jobs have inputs of 64MB–74GB and outputs of up to 4GB. These characteristics lead to an average datacenter utilization of 27%.

Nutch is the indexing part of the Web search system in CloudSuite [13]. The day-long trace consists of 2000 jobs that index groups of pages previously fetched from our Web domain. Each job runs 42 map tasks and 1 reduce task. Each map phase takes 15–40 seconds, whereas the reduce phase takes 150 seconds. On average, each job touches 85MB of data. Jobs arrive according to a Poisson distribution with mean inter-arrival time of 40 seconds. These characteristics lead to an average utilization of 32%.

We study both non-deferrable and deferrable (all jobs have 6-hour start deadlines) versions of these workloads.

Simulation and validation. Though we built CoolAir for real datacenters, we need to use simulations for multiple reasons, including (1) we cannot compare two real executions, because the same weather conditions never repeat exactly; (2) we want to run CoolAir multiple times, each time for an entire year, which is not feasible in real time; and (3) we want to study many geographical locations.

Thus, we built two simulators: Real-Sim and Smooth-Sim. *Real-Sim* simulates Hadoop on Parasol with or without CoolAir. Figure 6 compares a real execution of the baseline system (a) and the Real-Sim execution (b) on 07/02/2013 for the Facebook workload. Similarly, Figure 7 compares a real execution of CoolAir (b) and the Real-Sim execution (c) on 06/15/2013 for this workload (a). On this day, CoolAir tried to keep temperatures between 24 and 29°C. It is clear from the figures that Real-Sim is accurate. For the baseline system, maximum temperatures, temperature variations, and cooling energy are all within 8% of the real execution. For CoolAir, these values are within 15% of the real execution. In absolute terms, 89% of all real baseline measurements are within 2°C of its simulation, while 70% of the CoolAir measurements are within 2°C of its simulation. Overall, *we validate Real-Sim for 8 full random days,* with average simulation errors of 3.3% (baseline) and 6.6% (CoolAir).

Figure 7(b) illustrates a problem with Parasol's cooling infrastructure: it reacts too abruptly to changes in regime,

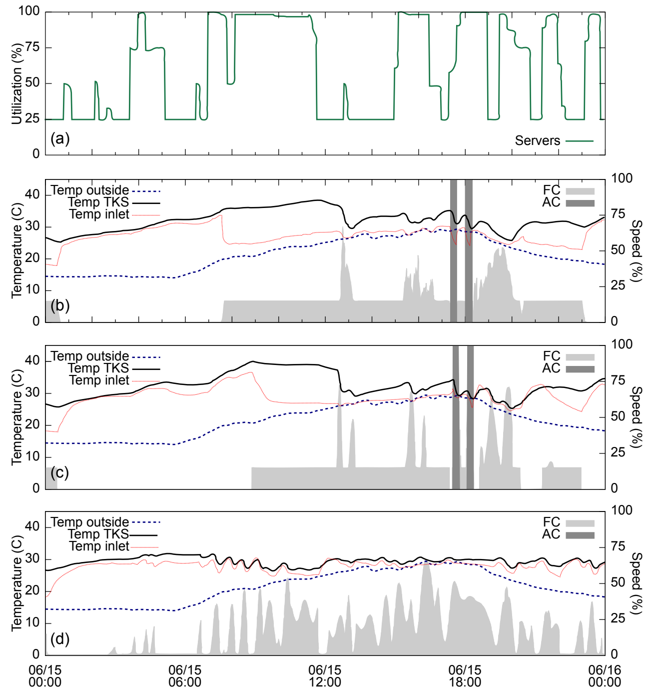

Figure 7. Workload (a), real (b), Real-Sim (c), and Smooth-Sim (d) CoolAir runs on 6/15/13.

making it difficult to manage variations accurately. For example, opening up Parasol to allow outside air to come in (causing free cooling to start running at 15% of the maximum speed) around 7:30am caused the inlet air temperature to decrease 9°C in only 12 minutes. Similarly, shutting down free cooling and turning on the AC (causing the compressor to run full-blast) caused a decrease of 7°C in 10 minutes.

Due to this limitation of Parasol, *Smooth-Sim* simulates CoolAir and Hadoop for a version of Parasol with a smoother, more controllable cooling infrastructure. Specifically, we simulate (1) a free cooling unit with fine-grained fan speed ramp up starting from 1% fan speed (ramp down still goes from 15% directly to off); and (2) an AC with fixed fan speed (ramp up is also fine-grained from 1% and settling at 100%), and variable and fine-grained compressor speed. Both the AC fan and compressor go straight from 15% to 0% when shutting down. These types of cooling units are available commercially [7, 11, 37]. Figure 7(d) shows the CoolAir behavior with this smoother infrastructure. Clearly, CoolAir keeps temperatures more stable with the new infrastructure.

Smooth-Sim cannot be validated in full, since we do not have access to a real fine-grained cooling infrastructure. However, we expect its behavior to also be close to reality, since the two simulators are almost exactly the same. To compute temperatures and humidity over time, they repeatedly call the same code implementing CoolAir's Cooling Predictor. We model the temperature, humidity, and power consumption of the smooth free cooling unit by extrapolating the earlier models to lower speeds. We model the temperature and humidity of the smooth AC by interpolating the models for the

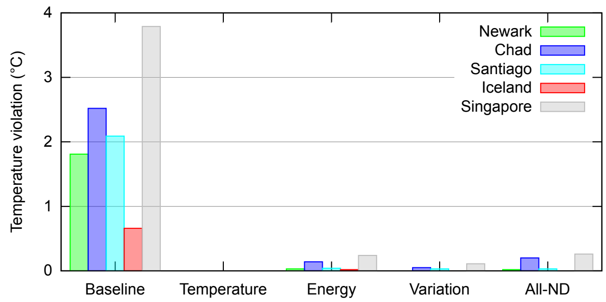

Figure 8. Average temperature violations.

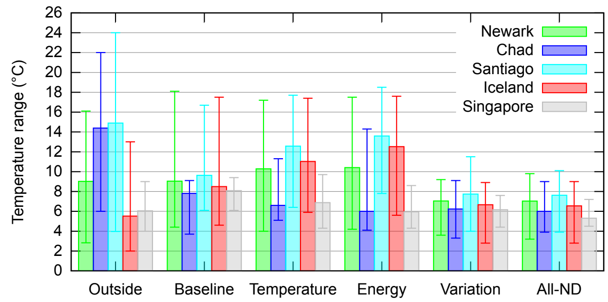

Figure 9. Temperature ranges.

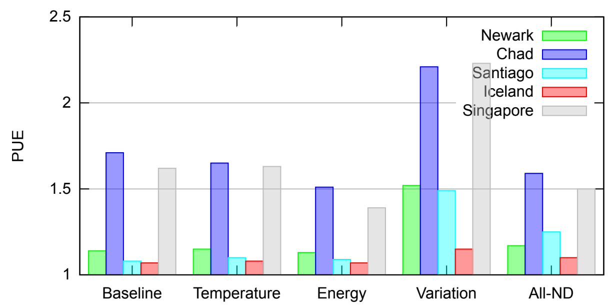

Figure 10. PUEs (including 0.08 for power delivery).

AC with the compressor on and off. For the power model, we assume the air conditioning fan consumes 1/4 of the power of the entire unit, and that the compressor consumes power linearly with speed. We base our assumptions, extrapolations, and interpolations in Smooth-Sim on [26].

To limit the length of our year-long Smooth-Sim simulations, we only simulate the first day of each week of the year. We repeat the workload for each of those days. We collect the typical meteorological year (TMY) temperature and humidity data for those days from [38]. We have compared our results for TMY and for actual temperatures for 2012 at two locations and found similar behaviors. Newark is the closest location to Parasol for which we found TMY data.

5.2 Results

Absolute internal temperatures. Figure 8 shows the average temperature "violations", *i.e.* the number of °C by which the systems exceed the desired maximum absolute air temperature of 30°C, during a year of running the non-deferrable Facebook workload at the five locations we consider. We compute this average by considering all sensor readings at or below 30°C to be a violation of 0°C. Each reading above 30°C contributes $reading - 30$ degrees to the overall sum.

These results show that the baseline system cannot limit all absolute temperatures well at warmer locations, especially in Singapore. As the other systems manage every sensor, they are more successful, with average violations lower than 0.5°C in all cases. Note that the bars for the Temperature version are *not* missing; this version is always able to keep average temperatures below 30°C (we set 29°C as the setpoint for Temperature at these locations) for the same PUE as the baseline system. All-ND is not as good as Temperature, but still leads to very low average violations. In fact, the CoolAir results are so good that we do not consider violations further.

Internal temperature variations. We measure the daily variation for each sensor as the difference between its maximum and minimum readings. From these variations, we select the worst sensor variation for each day. In Figure 9, the bars depict the average of these worst daily ranges over the year, while the vertical lines connect the minimum worst daily range to the maximum worst daily range. We also show the outside temperature variations (measured the same way).

These results show that the baseline system exhibits wide daily temperature ranges. Its average daily ranges hover around 9°C, and its maximum daily ranges are even wider. The maximum ranges are important because they represent an upper-bound on how variable a system is. For example, the maximum daily range for locations with cold or cool seasons (Newark, Santiago, and Iceland) is at least 16.5°C. The Temperature and Energy versions make maximum ranges even worse in some cases. As one would expect, Variation and All-ND lower the average daily ranges consistently, with respect to the baseline system. Most importantly, they lower the maximum daily range by a significant amount, except for Chad where it remains the same. For example, All-ND cuts the maximum daily range in *half* for Iceland, and *almost half* for Newark and Santiago. The most challenging days for All-ND are those when the outside temperature exhibits little overlap with CoolAir's temperature band.

Interestingly, the average outside temperature ranges can be lower than those inside under the baseline system (Iceland and Singapore). The same is the case for maximum ranges (Newark and Iceland). We mention the potential reasons for such behaviors in the Introduction.

PUEs. Figure 10 shows the yearly PUEs, assuming the power delivery losses of Parasol (0.08 in terms of PUE).

As one would expect, the baseline system exhibits high PUEs in Chad and Singapore. The Energy version reduces PUEs significantly at those locations. Interestingly, the Variation results show that trying to limit temperature ranges has a significant cooling energy penalty. All-ND brings PUEs back down to nearly the same values as the Energy version, except

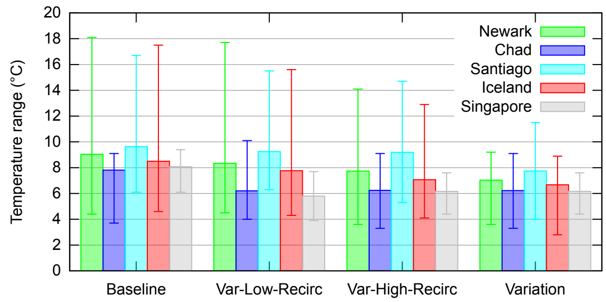

Figure 11. Temperature ranges, as a function of spatial placement and approach for limiting variation.

for Santiago. In this location, the baseline system does not need to consume much energy to keep temperatures below 30°C, whereas CoolAir does so to limit variation. However, two factors contribute to this: the relatively high power draw of air conditioning in our setup, and the low average IT power (servers are sent to sleep when not needed).

Cost of managing temperature and variation. We now quantify the yearly (energy) cost of controlling these measures. Lowering 1°C of absolute temperature costs more than reducing 1°C of maximum daily range in Newark (232 vs 53kWh), Chad (1275 vs 131kWh), and Singapore (2145 vs 716kWh). In Santiago (110 vs 171kWh) and Iceland (7 vs 29kWh), the opposite is true. Clearly, managing absolute temperature costs more than managing variation in places with warmer seasons, and less in places with colder ones.

Spatial placement. To understand why CoolAir achieves such large reductions in variation, we plot Figure 11. The figure depicts the temperature ranges for the baseline system, the Variation version of CoolAir, and two other systems: *Var-Low-Recirc* and *Var-High-Recirc*. Var-Low-Recirc attempts to keep temperatures between 25 and 30°C and uses a spatial placement that selects the pods with the lowest recirculation first. This placement has been used in prior works on energy-aware cooling in non-free-cooled datacenters [30, 32]. Var-High-Recirc tries to keep the temperature in the same range, but uses the same spatial placement as Variation. Var-High-Recirc uses no temperature band or weather prediction.

Comparing Var-Low-Recirc and Var-High-Recirc isolates the impact of the spatial placement. As the figure shows, placing loads on pods with higher recirculation reduces maximum temperature ranges somewhat; this tends to keep those areas consistently warm, with low variation. This is interesting because this spatial placement is the *opposite* as prior research has identified as ideal for energy savings [30, 32]. Nevertheless, high-recirculation placement increases PUE only slightly (not shown) in free-cooled datacenters.

However, the largest reductions in maximum range come from our temperature band (and the weather prediction it requires). Comparing Var-High-Recirc and Variation isolates its impact. As we can see, the maximum range goes down substantially for locations with cold or cool seasons when the band is used. The average ranges for these locations also go down but by smaller amounts.

Temporal scheduling. Now we consider whether the ability to defer workloads would enable reductions in temperature variation in All-DEF. Our results indicate that All-DEF provides only minor reductions in both average range and maximum range, compared to All-ND. The reason is that the days when All-ND does poorly in terms of variation are exactly those days when All-DEF decides to forgo temporal scheduling; All-DEF would perform worse if it tried to schedule for those days. For this reason, we argue that All-ND is the best implementation of CoolAir.

Temporal scheduling has been proposed recently for conserving cooling energy in free-cooled datacenters [2, 22, 27]. Though these techniques do conserve energy, *they also widen temperature variations*, compared to All-ND. We observe this by studying *Energy-DEF*, a system that combines the Energy version with temporal scheduling based solely on cooling energy. Like previous techniques, Energy-DEF schedules loads for periods when the outside temperature is low, but still within the jobs' start deadlines. As a sampling of the results, we find that the maximum range for Newark grows from 10 (All-ND) to 19°C (Energy-DEF), in exchange for a reduction in PUE from 1.17 to 1.13. For Santiago, the maximum range grows from 10 to 18°C, while PUE decreases from 1.25 to 1.10. For all five locations, the Energy-DEF maximum ranges are even worse than those of the baseline system.

Impact of the desired maximum temperature. Now we investigate the impact of trying to keep absolute temperatures below different maximum values. For the baseline system, this means setting the temperature setpoint to those values. For CoolAir, this means not allowing the top of the adaptive temperature band to exceed those values, which can be done by setting *Max* to each of them.

We find that the CoolAir benefits tend to be greater when datacenter operators are willing to accept higher maximum temperatures (to allow for lower cooling costs). For example, the reductions in maximum range achieved by CoolAir tend to be greater for a desired maximum temperature of 30°C than 25°C. For locations where PUE is high for a desired maximum temperature of 30°C, CoolAir tends to lower PUEs. However, CoolAir tends to increase PUEs for those same locations when the desired maximum temperature is 25°C.

Impact of weather forecast accuracy. As we only have the TMY data, our simulated predictions of average outside temperature are perfectly accurate. Fortunately, weather services predict daily average temperatures accurately. For example, at our location, predictions are within 2.5°C of the actual daily averages 83% of the time, *i.e.* plenty accurate given CoolAir's 5°C-wide temperature band.

Nevertheless, we next quantify the impact of inaccurate outside temperature predictions on CoolAir. Specifically, we consider scenarios in which average outside temperature predictions are consistently too high by 5°C, and consistently

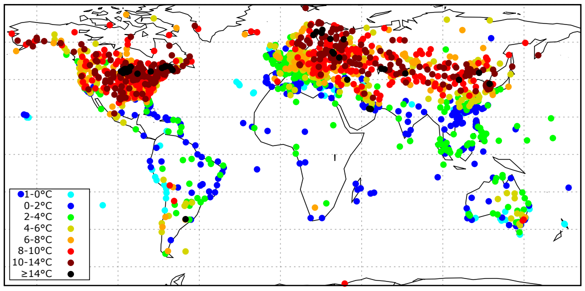

Figure 12. World-wide reduction in max range.

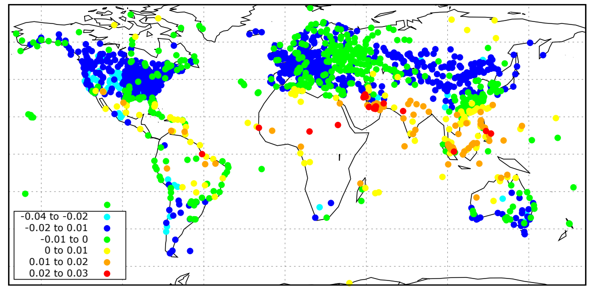

Figure 13. World-wide reduction in yearly PUE.

too low by 5°C. For the former case, our results show increases of the maximum ranges but always by less than 1°C, and reductions in PUE. For the latter case, our results show reductions in the maximum ranges, and increases in PUE but always by less than 0.01. Clearly, the impact of inaccuracies is small, mostly because of CoolAir's temperature band.

Impact of workload. The results above have been for the Facebook workload. We now study the widely different Nutch trace. The results with Nutch exhibit the exact same trends we observe with the Facebook workload (Figures 9 and 10). For example, All-ND cuts the maximum daily temperature range in roughly half for Newark, Santiago, and Iceland, while also lowering the average daily range for all locations. These benefits come with significant PUE reductions for Chad and Singapore, and a small PUE increase for Santiago.

More geographical locations. Finally, we now extend our study to 1520 locations world-wide, assuming the Facebook workload. We do not argue that datacenters could be built at all locations, since there are other issues to consider [16]; rather, we quantify the potential impact of using CoolAir at the locations were datacenters to be built there.

Compared to the baseline system, CoolAir reduces the maximum range from 18.6 to 12.1°C on average, for a slight increase in yearly PUE from 1.08 to 1.09 on average. In more detail, Figures 12 and 13 depict the reductions in maximum range and PUE, respectively, compared to the baseline system. The figures show that CoolAir can reduce maximum ranges significantly in colder locations. Specifically, it can reduce these ranges by between 2 and 14°C in a large number of locations in North America, Europe, and Asia. Similarly, it can reduce the maximum ranges by between 2 and 8°C in many locations in the south of South America and Australia. These reductions come with only a slight penalty in PUE. In fewer than 2% of locations, CoolAir increases the maximum range, but always by less than 1°C.

CoolAir is also useful for locations closer to the Equator, which exhibit higher PUEs. For those locations, CoolAir lowers PUEs without increasing internal temperature variations. Thus, CoolAir should be useful all over the world.

6. Principles and lessons

In summary, our results prompt many observations:

1. If not managed explicitly in free-cooled datacenters, absolute temperatures and variations can be high in many locations (Figures 8 and 9), especially when temperature setpoints are higher. In fact, internal variations can be higher than those outside (Figure 9). These effects have been shown to degrade disk reliability [10, 34, 36].

2. Effectively managing temperature variation requires fine-grain cooling (Figures 7(b) and (d)) and workload control (Figures 9 and 11).

3. Managing absolute temperature costs more than managing variation in regions with warmer seasons, and less in regions with colder ones (Cost of managing temperature and variation, Section 5.2).

4. Adaptive temperature bands and smart spatial workload placement are useful in managing temperature variation (Figure 11), whereas temporal workload scheduling is not (Temporal scheduling, Section 5.2). Moreover, existing energy-driven techniques for spatial and temporal thermal management increase temperature variation (Figure 11 and Temporal scheduling, Section 5.2).

5. Managing absolute temperature and temperature variation is easier when internal temperatures can be higher (Impact of the desired maximum temperature, Section 5.2).

6. The accuracy of weather forecast services is not a problem when managing variations with temperature bands (Impact of weather forecast accuracy, Section 5.2).

7. Managing temperature variation is most critical and successful in cold climates (Figures 12 and 13).

8. Using CoolAir, it is possible to manage absolute temperature and variation at little or no energy cost even in very hot climates (Figures 12 and 13). *This significantly broadens the set of locations where free cooling can be used.*

Generalizing from the case study. Clearly, the results of Section 5.2 are influenced by the characteristics of Parasol. However, we expect that *the observations above should apply to any datacenter,* as our modeling of air temperature and cooling power is similar to that of very different datacenters in the literature. Specifically, the model we learned for air temperature is similar to that in [20]. The power model for free cooling models power as a cubic function of fan speed, as in [27]. Our AC power model was derived experimentally in [26]. For datacenters that combine free cooling with chillers (instead of DX AC), we can use [23] to strike the proper ratio of power consumptions. For a large datacenter with multiple independent "cooling zones" (*e.g.*, containers), each of them would have its own CoolAir-like manager.

Thus, applying our ideas to another datacenter would lead to different temperatures, variations, and PUEs, but the trends (*e.g.*, impact of climate on variation) should be the same.

Practical considerations. CoolAir can be used in a variety of real scenarios, from small-scale datacenters found in universities to large-scale datacenters operated by Internet companies. Free-cooled datacenters, especially large-scale ones, embody temperature, humidity, and power sensors. These sensors facilitate the creation of the corresponding CoolAir models over time (*e.g.*, 6 months or 1 year), *during the normal operation of the datacenter.* (We explicitly generated scenarios for Parasol simply to speedup our learning of the temperature and humidity models.)

Selecting the CoolAir *Offset* and *Width* parameters can be easily done via observation and experimentation, respectively, for a desired tradeoff between ambient control and energy consumption. In fact, our simulation infrastructure would allow the datacenter operator to evaluate multiple settings even before real deployment.

Finally, many real systems and workloads enable flexibility in spatial and temporal scheduling. Spatial flexibility needs to exist for fault tolerance and performance optimization, whereas temporal flexibility exists in real deferrable workloads (*e.g.*, typical batch and data-processing loads).

7. Related Work

To our knowledge, this paper is the *first publicly available study* of the internal absolute temperatures and temperature variations in a real free-cooled datacenter. Though companies like Google and Facebook use free cooling [14, 18], they have not published data on temperature or humidity effects in these datacenters or their disk reliability. Our paper is also the *first study* of techniques for managing temperature and variation in free-cooled datacenters. Finally, our paper is the *first to argue* for (1) fine-grained control of the cooling infrastructure for managing variation; and (2) selecting temperature setpoints based on predictions of outside temperature. Nevertheless, CoolAir does relate to works on the following three topics.

Disk reliability in datacenters. Our motivation to manage absolute temperature and temporal temperature variation in free-cooled datacenters came from papers on disk reliability in non-free-cooled datacenters [10, 34, 36].

Energy-aware thermal management of non-free-cooled datacenters. Similarly to CoolAir, previous efforts have used: machine learning for modeling thermals (*e.g.*, [25, 31]); programmatic control of the cooling infrastructure (*e.g.*, [6, 40]); simulations to predict thermal behavior into the future (*e.g.*, [35]); energy-aware spatial placement of workloads (*e.g.*, [1, 4, 5, 30, 32]); and energy-aware temporal scheduling of workloads (*e.g.*, [4, 32]). The CoolAir contributions here are simulator-based predictions, *spatial placement, and temporal scheduling that seek to manage temperature variation primarily,* not cooling energy. In fact, our results show that the best spatial placement for variation is the *opposite* of the best placement for energy.

Energy-efficiency in free-cooled datacenters. Researchers have studied free cooling and its impact on energy efficiency (*e.g.*, [15, 39]); programmatic control of it (*e.g.*, [12, 41]); techniques for temporal scheduling based on outside temperatures (*e.g.*, [2, 22, 27]); and geographical load balancing based on outside temperatures (*e.g.*, [23]). However, as mentioned above, *none of these studies* considered internal temperature variations, selecting setpoints based on predicted outside temperatures, or techniques for managing variations. In fact, our results demonstrate that temporal scheduling for energy [2, 22, 27] *increases* temperature variations significantly compared to CoolAir.

8. Conclusions

We designed, implemented, and evaluated CoolAir, a system for managing temperature, variation, humidity, and energy in free-cooled datacenters. Though our case-study results are based on Parasol, a real free-cooled datacenter prototype, we derived lessons that apply to other datacenters as well. Importantly, these lessons are useful regardless of how researchers eventually resolve the issue of whether absolute temperature or temperature variation has the strongest impact on hardware reliability. As CoolAir shows, it is possible to manage *both* effects while keeping cooling energy consumption low.

Acknowledgments

We would like to thank our shepherd, Hillery Hunter, as well as Georgios Varsamopoulos, Cristian Perfumo, and Sriram Sankar for comments that helped improve our paper. We are also indebted to NSF grant CSR-1117368 and the Rutgers Green Computing Initiative for their funding of our work.

References

[1] F. Ahmad and T. Vijaykumar. Joint Optimization of Idle and Cooling Power in Data Centers while Maintaining Response Time. In *ASPLOS*, 2010.

[2] M. Arlitt, C. Bash, S. Blagodurov, Y. Chen, T. Christian, D. Gmach, C. Hyser, N. Kumari, Z. Liu, M. Marwah, A. McReynolds, C. Patel, A. Shah, Z. Wang, and R. Zhou. Towards the Design and Operation of Net-zero Energy Data Centers. In *ITherm*, 2012.

[3] ASHRAE Technical Committee. 2011 thermal guidelines for data processing environments – expanded data center classes and usage guidance. *Whitepaper*, 2011.

[4] A. Banerjee, T. Mukherjee, G. Varsamopoulos, and S. Gupta. Cooling-Aware and Thermal-Aware Workload Placement for Green HPC Data Centers. In *IGCC*, 2010.

[5] C. Bash and G. Forman. Cool Job Allocation: Measuring the Power Savings of Placing Jobs at Cooling-Efficient Locations in the Data Center. In *USENIX ATC*, 2007.

[6] C. Bash, C. Patel, and R. Sharma. Dynamic Thermal Management of Air Cooled Data Centers. In *ITherm*, 2006.

[7] Carbon Trust. Variable Speed Drives, 2013. `http://www.carbontrust.com`.

[8] Y. Chen, A. Ganapathi, R. Griffith, and R. Katz. The Case for Evaluating MapReduce Performance Using Workload Suites. In *MASCOTS*, 2011.

[9] Y. Chen, S. Alspaugh, A. Ganapathi, R. Griffith, and R. Katz. Statistical Workload Injector for MapReduce, 2012. `https://github.com/SWIMProjectUCB/SWIM`.

[10] N. El-Sayed, I. Stefanovici, G. Amvrosiadis, A. Hwang, and B. Schroeder. Temperature Management in Data Centers: Why Some (Might) Like it Hot. In *SIGMETRICS*, 2012.

[11] Emerson Network Power. Liebert DSE – DX Cooling and Refrigerant Economizer System, 2013. `http://www.emersonnetworkpower.com`.

[12] H. Endo, H. Kodama, H. Fukuda, T. Sugimoto, T. Horie, and M. Kondo. Effect of Climatic Conditions on Energy Consumption in Direct Fresh-Air Container Data Centers. In *IGCC*, 2013.

[13] EPFL. CloudSuite, 2012. `http://parsa.epfl.ch/cloudsuite/`.

[14] Facebook. Prineville - Facebook Power, 2013. `https://fbpuewue.com/prineville`.

[15] B. Gebrehiwot, K. Aurangabadkar, N. Kannan, D. Agonafer, D. Sivanandan, and M. Hendrix. CFD Analysis of Free Cooling of Modular Data Centers. In *SEMI-THERM*, 2012.

[16] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini. Intelligent Placement of Datacenters for Internet Services. In *ICDCS*, 2011.

[17] I. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini. Parasol and GreenSwitch: Managing Datacenters Powered by Renewable Energy. In *ASPLOS*, 2013.

[18] Google. Efficiency: How We Do It - Data Centers - Google, 2013. `http://www.google.com/about/datacenters/efficiency/internal/`.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1), 2009.

[20] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini. Mercury and Freon: Temperature Emulation and Management for Server Systems. In *ASPLOS*, 2006.

[21] Hitachi. Ramp Load/Unload Technology in Hard Disk Drives, 2007. `http://www.hgst.com`.

[22] N. Kumari, A. Shah, C. Bash, Y. Chen, Z. Liu, Z. Wang, T. Cader, M. Slaby, D. Cepulis, C. Felix, A. Aviles, and M. Figueroa. Optimizing Data Center Energy Efficiency via Ambient-Aware IT Workload Scheduling. In *ITherm*, 2012.

[23] K. Le, J. Zhang, J. Meng, R. Bianchini, Y. Jaluria, and T. D. Nguyen. Reducing Electricity Cost Through Virtual Machine Placement in High Performance Computing Clouds. In *SC*, 2011.

[24] J. Leverich and C. Kozyrakis. On the Energy (In)efficiency of Hadoop Clusters. In *HotPower*, 2009.

[25] L. Li, C. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos. Thermocast: A Cyber-Physical Forecasting Model for Data Centers. In *KDD*, 2011.

[26] Z. Li and S. Deng. An Experimental Study on the Inherent Operational Characteristics of a Direct Expansion (DX) Air Conditioning (A/C) Unit. *Building and Environment*, 42, 2007.

[27] 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.

[28] R. Miller. Free Cooling, Illustrated, 2009. `http://www.datacenterknowledge.com/archives/2009/02/03/free-cooling-illustrated/`.

[29] R. Miller. The Evolution of Facebook's Data Center Cooling, 2012. `http://www.datacenterknowledge.com/archives/2012/12/04/evolution-of-facebooks-cooling/`.

[30] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making Scheduling "Cool": Temperature-Aware Workload Placement in Data Centers. In *USENIX ATC*, 2005.

[31] J. Moore, J. Chase, and P. Ranganathan. Weatherman: Automated, Online and Predictive Thermal Mapping and Management for Data Centers. In *ICAC*, 2006.

[32] T. Mukherjee, A. Banerjee, G. Varsamopoulos, S. Gupta, and S. Rungta. Spatio-Temporal Thermal-aware Job Scheduling to Minimize Energy Consumption in Virtualized Heterogeneous Data Centers. *Computer Networks*, 53(17), 2009.

[33] J. Park. Open Compute Project–Data Center. Technical report, Facebook, 2011.

[34] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure Trends in a Large Disk Drive Population. In *FAST*, 2007.

[35] L. Ramos and R. Bianchini. C-Oracle: Predictive Thermal Management for Data Centers. In *HPCA*, 2008.

[36] S. Sankar, M. Shaw, K. Vaid, and S. Gurumurthi. Datacenter Scale Evaluation of the Impact of Temperature on Hard Disk Drive Failures. *ACM Transactions on Storage*, 9(2), 2013.

[37] Stulz. Stulz Solutions and Services, 2013. `http://www.stulz-ats.com`.

[38] US DOE. Weather Data. `http://apps1.eere.energy.gov/buildings/energyplus/weatherdata_about.cfm`.

[39] B. Weerts, D. Gallaher, R. Weaver, and O. V. Geet. Green Data Center Cooling: Achieving 90% Reduction: Airside Economization and Unique Indirect Evaporative Cooling. In *GreenTech*, 2012.

[40] R. Zhou, Z. Wang, C. B. E, A. McReynolds, C. Hoover, R. Shih, N. Kumari, and R. Sharma. A Holistic and Optimal Approach for Data Center Cooling Management. In *ACC*, 2011.

[41] R. Zhou, Z. Wang, A. McReynolds, C. Bash, T. Christian, and R. Shih. Optimization and Control of Cooling Microgrids for Data Centers. In *ITherm*, 2012.