Detecting Data Center Cooling Problems Using a Data-driven Approach

Charley Chen, Guosai Wang, Jiao Sun, Wei Xu
Institute for Interdisciplinary Information Sciences
Tsinghua University, Beijing 100084, China.
{ctj2015,weixu}@mail.tsinghua.edu.cn,{wgs14,j-sun16}@mails.tsinghua.edu.cn

ABSTRACT

Cooling problems are common in data centers and many of them are hard to detect especially the hidden. These problems affect overall system dependability, performance and power efficiency. We propose a novel method to detect the cooling problems. Using common monitoring data available in most data centers, such as environmental temperature and hardware status, we build a workload-independent cooling profile for each server. With the cooling profiles, we are able to detect two types of both transient and lasting cooling failures. We detect transient failures by comparing the observed temperature with the model prediction, while we detect lasting failures by comparing the cooling profiles among different servers. We demonstrate the general applicability of our detection methods in three production data centers with vastly different scale, server types and workload, and detect several real cooling problems that have been hidden for months.

CCS CONCEPTS

•Datacenter Operations → Cooling Reliability;

KEYWORDS

Cooling Profile, Data-driven approach, Abnormal detection

1 INTRODUCTION

Cooling problems happen frequently in data centers. In the early days, Jim Gray [1] and David Patterson [2] survey summarizes that about 32% of the system errors are caused by hardware and cooling problems. For example, if the CPU temperature continues to increase, the catastrophic shutdown detectors will force processor to halt [6]. Cooling failures at this severity is usually associated with a critical situation in data centers and they often get immediate attention. While these kinds of failures are more eventful, they happen rarely and are easy to detect and fix.

Cooling problems do not necessarily lead to an observable event of equipment overheating. People have designed layers of hardware, software and operation procedures to tolerate cooling problems to improve system dependability. As a result, many cooling problems become quiet, but these hidden cooling failures can still lead to performance, dependability or power efficiency problems.

APSys'18, Jeju, South Korea

DOI: 10.475/123_4

For example, hard disks can operate at a relatively high temperature, but as [17] points out, the dependability of disks significantly degrade at a high temperature. Automatic fan control can speed up the cooling fan to alleviate problem of a partially blocked server air inlet. This action prevents immediate server overheating, but increases the power draw on the fan. Even worse, a common practice to avoid cooling problem is to reduce the room temperature to ensure a safe margin, and the margin is usually set to tolerate the equipment with the worst cooling performance. For example, we try to reduce energy consumption by turning off one air conditioner. Then hidden cooling problems appears, one server with Titan X GPU turn to be extremely hot and other two servers cannot boot up, these servers fail because the GPU driver version does not match the server's kernel, so it cannot control the GPU fan correctly, the other two have bad mother boards that fails when temperature goes high.

Thus, we believe it is essential to detect cooling problems, especially those hidden problems, in a robust, low overhead and fast way, in addition to tolerate them by cooling more oppressively, and it is the goal of this paper. We mainly focus on servers, the largest type of heat generating devices in data centers, but our approach is applicable to other types of equipments too.

Many factors contribute to an abnormal temperature of a server. There are the three major categories, including the server's own hardware/software, its environment and the workload [7]. There are many reasons for each factor to go wrong. We see certain corners in our data center that do not receive enough chilled air, creating a bad environment for whichever server placed there. Human mistakes may lead to some hard to detect errors. For example, we see a situation that the human operator accidentally left a gap between floor tiles, which let some chilled air leak out, causing cooling problems for a few racks. We present more of such real world problems in our evaluation section.

A lot of existing work focus on cooling air flows in the data centers [19],[20],[12]. However, air flow is particularly difficult to model, especially in small scale data centers, as the limited physical space causes lots of non-linear and fast changing behavior in cooling air circulation [8]. Cooling provisioning at the design time do not work well as administrators constantly adds/removes equipments, making airflow circulation even less predictable.

Instead of directly measuring the airflow, we take a data driven approach: we monitor the common metrics of a data center, such as the server utilization, temperature of the server components, and build a statistical model to predict each server's cooling behavior under *any* given workload and current thermal state. We call the model *cooling profile* of the server. The system learns the cooling profile based on the servers' cooling behavior on various workload levels, but after the server's cooling profile was built, we can use it to detect problems independent of its current workload. Workload-independence is the key to our approach: it allows us to detect hidden cooling issues even if there is no observable overheating.

The key challenge of obtaining the cooling profile from a production data center is that the servers are running different workloads and it would be too expensive to stop all production job to perform a system identification for thermal modeling. We take a data-driven approach and collect monitoring data from servers under production workload, which are very noisy, for a relatively long period of time (a couple of days), and then we can build statistical model for each individual server.

We use the cooling profiles to detect two different kinds of cooling failures: *transient failures* and *lasting failures*.

It appears more straightforward to detect transient failures than it actually is. Many available tools use a static temperature threshold. It is hard to set the threshold accurately because it is workload-dependent and some servers can get hotter than others due to their configuration or environment. Lasting failures are even harder to detect. Some servers have a poor cooling behavior to begin with because of non-fatal hardware / software bugs or poor locations in the data center. Our cooling profile captures its bad behavior as the server's "normal" condition and compare a server's cooling profile with that of other servers. Specifically, we define a distance metric between any two cooling profiles and apply machine learning algorithms such as clustering and anomaly detection to put each server into groups based on their cooling profile. Servers with unique cooling profile are suspects for having cooling problems.

Empirical experiments show that the detection is quite accurate. Note that our detection is independent of the *cause* of the cooling problem. Some non-circuit failures, such as server fans cover for the misplaced server , is hard to find as there is no special sensors for that. Our mechanism can successfully detect these problems.

To demonstrate the wide applicability of our approach, we perform experiments in three data centers with different scales and hardware/software configurations details in 4.1.

We only use the common metrics such as workload and temperature that are readily available. We show that for all three data centers, we can not only quickly alert about transient failures, but we can also find many lasting failures that have remained unnoticed for months in the production data center with advanced thermal monitoring system.

In summary, we made the following contributions:

(1) We propose a novel model, the *cooling profile* to capture the intrinsic cooling behavior of a server that is independent of workload.
(2) We design a machine-learning based approach to detect both transient and lasting cooling problems.
(3) We develop a data driven approach that allows us to build the cooling profile without any disturbance to the production data center.
(4) We applied our approach in three distinct data centers and found many real world cooling problems.

The remaining of the paper is organized as the following. Section 2 reviews the related work. We formulate the problem and provide a general introduction on our approach, discusses the details of our thermal modeling and anomaly detection methods in Section 3. In Section 4, we present our evaluation result, and finally we conclude in Section 5.

2 RELATED WORK

Data center cooling is an important topic related to both system dependability and energy efficiency. For dependability, people have studied correlations between equipment's thermal environment and component failures. For example, People have analyzed the influence of temperature on disk failures [16][17], memory reliability [18] and energy consumption [7]. People have also pointed out the impact of thermal environment on the server performance [7].

Researchers have designed many approaches to improve data center cooling. There are two major methods: optimizing the air flows in the data center and optimizing the job placement with in the data center. While the airflow modeling is mostly related to thermal management only, the job placement and scheduling can help both thermal control and data center power control.

2.1 Optimizing data center airflows

Abnormal air flow is the worst enemy for energy efficiency. A large portion of failures are related to the abnormal air flows [19],[12],[42],[43]. Bad air flow pattern makes servers overheat, affecting both performance and dependability. Researchers have used *Computational Fluid Dynamics (CFD)* to model air flow and heat transfer [10, 11]. Researchers combine computational fluid dynamics modeling and real-time data-driven prediction algorithms [13], leveraging the accuracy of CFD and the real sensor measurements to achieve high fidelity temperature forecasting. Paper [14] presents an sensor placement method to detect the thermal abnormal server models the airflow within a server to improve measurement accuracy. They are typically used to design or redesign large-scale data centers.

2.2 Thermo aware job placement in a data center

Job scheduling is a complex task in data centers. Schedulers need to consider many factors, such as computation resources, networking as well as power distribution. [27, 23, 28, 29] combine heat transfer properties and workloads to provide thermal-aware scheduling method for the distributed computing servers, to reduce the data center operational costs while guaranteeing the service level. People have developed many thermal-aware workload management mechanisms to save power [21, 22, 24, 25]. Paper [33] uses renewable energy such us solar-cell, and schedule more work during the day when renewable energy is available.

2.3 Data driven data center cooling and power modeling

Researchers at Google have implemented neural networks to model the power utilization efficiency of their large-scale data centers [30]. Using a statistical model, they can study the influence of one or more controllable parameters on the power efficiency. The data center thermal topology can also be derived by neural network model [31]. The thermal topology is useful in tasks such

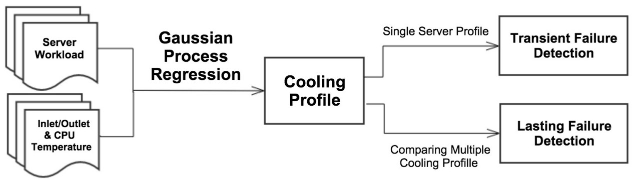

Figure 1: Architecture Of Our Approach

as enhancing hardware reliability, reducing cooling cost, shortening the reaction time of vital failures. These models all depend on current workload and focus on the global states of the date center. The work [26] provides a comprehensive analysis on how cooling infrastructures impact data center's sustainability, cost and dependability. The authors present five real-world data center cooling architectures and data to explore the environmental impact and dependability metrics.

3 SYSTEM DESIGN

In this section, we provide a big picture of our design and overview of system architecture. Fig. 1 illustrates the overview of our system for cooling problem detection. The system consists of five major components and will be explained one by one in this section.

3.1 Design choices

Our goal is to perform fast and automatic detections on both transient and lasting cooling failures in data centers. The detection is on a local (server or rack) scale. We made the following two design choices as the basis for various trade-offs in our design.

1) We make our model flexible and robust to accommodate existing workload and thermal data collected in the data centers, instead of the other way around (e.g. tracking air flow). In other words, we want our model be generally applied to in different types of data centers.

2) Instead of using complex data and models to reason about the root causes of the cooling problems, we focus on quickly and accurately detection for the hidden cooling problems with a generally applicable model. Our experience shows that unlike software bugs, for cooling problems, once the operators know where exactly to look at, they can quickly determine the root cause.

3.2 Data Collection.

The temperature readings of servers are the most crucial indicator of the cooling condition. However as we have discussed in Section 1, we cannot only use a static threshold to distinguish normal from abnormal.

The key observation of this work is that server workload, especially the CPU utilization, strongly affects the temperature readings, but the cooling failures is intrinsic to the server and its environment, not the workload. In our data center, the normal CPU temperature can range from 38°C to 56°C under different workload. In other words, servers get hot anyways when the CPU utilization increases, and what problematic is whether the temperature is getting too high or increasing too fast only because of a certain workload. A naive approach is to model the workload directly, but it does not work because modern cloud computing data centers run heterogeneous workload which is not very predictable. For example, one server with GPU, the CPU temperature situation mainly depends on the GPU's fan, with any workload the CPU temperature maintains around 25 °C.

Some metrics, such as the CPU utilization, changes very quickly, so the metrics may not be collected at the same time interval. In order to build a consistent model, we first pre-process the data by putting them into the same time scale. We also smooth metrics like CPU utilization using exponential moving average.

$$W^{EMA} = \begin{cases} W_1 = W_{obsv} & (Time = 1) \\ W_2 = decay * W_{obsv} + (1 - decay) * W_{obsv-1} & (Time = 2) \\ W_3 = decay^2 * W_{obsv} + (1 - decay) * decay * W_{obsv-1} & \\ \qquad +(1 - decay) * W_{obsv-2} & (Time \geq 3) \end{cases}$$

Where W_{obsv} is the current CPU utilization, W_{obsv-1} is the last time CPU utilization, decay is the decay factor. Intuitively, the moving average function includes a small amount of history data into the model, while still let the most recent value to affect the model outcome the most.

For data collection, we focus on using monitoring data already available in different DC (Data Center information will be shown in Section 4.1) . DC-C already has a large scale monitoring infrastructure collecting 83 metrics every minute from the entire fleet on workload. For the other two DCs we collect workload data from the operating systems and the temperature data from the baseboard management controller (BMC) through the standard Intelligent Platform Management Interface (IPMI) [37] interface. The period for data collection varies from DC to DC too. We take a reading from DC-A and DC-B every 5 seconds, while DC-C only report a reading every 1 minute.

3.3 Cooling profiles

After data collection, we define a workload-independent cooling profile. The cooling profile is a model that predicts the next temperature reading given the current temperature and the workload. Both internal factors (e.g. power consumption, cpu frequency, fan speed) and external factors (e.g. environment temperature) contribute to the cooling profile and it independent to the current task and workload. More formally, we define the cooling profile as a function $\mathbf{\Phi}$ from the domain ($\mathbf{T_0}$, $\mathbf{W}$) to the range of the server temperature $\mathbf{T}$

$$\mathbf{\Phi} : (\mathbf{T_0}, \mathbf{W},) \rightarrow \mathbf{T} \tag{1}$$

where $\mathbf{T_0}$ represents the current temperature (e.g. CPU, inlet/outlet temperature) and $\mathbf{W}$ represents the workload. The output $\mathbf{T}$ is the prediction CPU temperature. To capture the potential inaccuracy of the prediction, we treat the output $\mathbf{T}$ as a random variable following a specific probability distribution. For example, if we assume $\mathbf{T}$ follows a Gaussian distribution, we can estimate that $\mathbf{T}$ should be within the range of $(\mu - 3\sigma, \mu + 3\sigma)$ with high probability, where μ and σ are the mean and the standard deviation of the distribution.

Note that we capture within the function $\mathbf{\Phi}$ all internal or external factors, other than the workload and temperature. As each server has some uniqueness in its cooling capability and location in the data center, $\mathbf{\Phi}$ can be different across different servers. Also, as $\mathbf{\Phi}$ captures multiple factors affecting cooling, it is likely to be non-linear. Our experiments confirm the fact.

It is impractical in a production data center to stop all task on servers to conduct experiment. We take a data driven approach

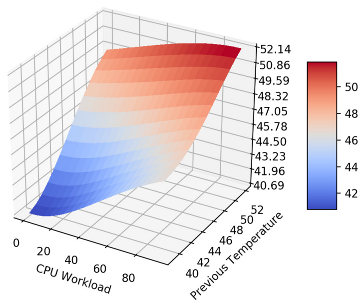

Figure 2: Typical cooling profile.

to evaluate Φ for each server. We observe workload and corresponding temperature readings from each server for a period of time. Intuitively, if the server runs variable workload, over time we will be able to observe the behavior of a server under different workload and at different temperature. Or course the observations are noisy and we tolerate the noise by taking more measurements. Then we use the observed data to statistically determine Φ.

Each of the dimensions can vary a lot. Theoretically, to capture the system cooling performance under all conditions, we need to observe all possible situations, which is impractical in our settings of collecting data under production workload, given the huge multi-dimensional domain space. Thus, we use a regression model to fit a function to capture the entire domain. In another word, we interpolate the missing spaces we have not seen in the observed history.

We choose *Gaussian Process Regression (GPR)* to construct the statistical model for the following two reasons.

Firstly, we observe that the server's temperature (output of Φ) follows Gaussian distribution, which fits the assumptions of the GPR model. This is not a coincidence. Many factors contribute to the server temperature. Considering these contributing factors random variables (dependent or independent), by central limit theorem the server temperature should have a normal distribution. The data we collected with high noise level, even with same environment conditions, CPU temperature have low probability rising/reducing by 1 °C. Depending on the initial value for our data sets, GPR can converge these noise based on the gradient-based optimization and repeat the optimization several times for different initializations.

Secondly, GPR is a relatively simple model that captures the non-linear relationship between $\mathbf{T_0}$, $\mathbf{W}$ and $\mathbf{T}$. We can consider a simple linear model to approximate the function, however, the linear model significantly reduces the model accuracy. Of course, more complex non-linear models such as conditional random fields or neural networks can achieve similar performance. However, we found that it is 20 times slower GPR model. Thus, we believe GPR is a good tradeoff between model accuracy and training performance.

Figure 2 shows a typical cooling profile built for a server in DC-B. For clarity, we only plot three dimensions: the exponential moving average workload, the previous temperature and the prediction temperature. The curve shows illustrates how fast the temperature will rise at each given temperature and workload. Intuitively, the more "flat" the curve is, the better cooling capacity a server has, as the temperature of the server rises slower at different workload.

3.4 Transient and lasting cooling failures

As we have discussed in Section 1, there are two kinds of cooling problems, transient and lasting. We provide a more formal definition of each kind based on our cooling profile definition.

The transient failure can be a overheating or under-heating. Though overheating is more of a risk to the overall dependability, it is also important to detect under-heating cases that often lead to waste of energy. For example, when a over-working fan sucking in too much chilled air, the overall cooling efficiency of its neighbers would suffer. To detect transient failures, we find situations in which the observed temperature deviates from the predicted value from the cooling profile. We raise an alert when the amount of deviation is above a automatically determined threshold.

Given an observed temperature value of $\mathbf{T_0}$ and output of Φ of $\mathbf{T}$, we define the *anomaly score* as

$$S = |\Phi - \mathbf{T_{obsv}}| / \mathbf{T_{obsv}} \tag{2}$$

Given a reasonably accuracy but not perfect cooling profile Φ, S should be a small value in normal cases. $S > threshold$ indicates an anomaly. Assuming most of the times the server stays normal (which is not true with a lasting failure), we can automatically determine the *threshold* for each server, independent of its workload.

Lasting failures are "bad" cooling profiles. That is, a function Φ itself is a poor mapping, indicating that the server temperature changes too fast too slow, or unstable in a bizarre ventilation passage. These problems can be detected once the cooling profiles are constructed. We use anomaly detection algorithm based on clustering to dig out such abnormal cooling profiles.

While it is hard to statically decide whether a cooling profile is normal or not, as there are many servers in a data center, we determine if a server has a normal cooling profile by comparing it to its peer servers. We firstly define a distance metric *cooling profile distance* between two cooling profiles, and then describe how to use anomaly detection algorithm to find abnormal cooling profiles based on the cooling profile similarity metric.

To compare the cooling profile of different servers, we discretely sample the values of Φ on a selected range in each dimension, which forms a high dimensional vector. We normalize all temperature values by subtracting the mean temperature of the particular server to provide a comparison that focus on the *shape* of the cooling profile instead of its absolute value.

For example, if the input domain of Φ has two dimensions x, y, and we assume discretely 2 and 3 values in a range of x and y, and the mean value of server temperatures of a certain server s is $\overline{T}$, then the high dimensional vector $\omega_{\mathbf{s}}$ we get from the server is

$$\begin{aligned} \omega_{\mathbf{s}} = (&\Phi(x_0, y_0) - \overline{T}, \Phi(x_0, y_1) - \overline{T}, \Phi(x_0, y_2) - \overline{T}, \\ &\Phi(x_1, y_0) - \overline{T}, \Phi(x_1, y_1) - \overline{T}, \Phi(x_1, y_2) - \overline{T})^T \end{aligned} \tag{3}$$

We define the cooling profile distance between two servers s_1 and s_2 as

$$S(s_1, s_2) = \|\omega_{\mathbf{s_1}} - \omega_{\mathbf{s_2}}\|_2 \tag{4}$$

which is the L2-Norm or the Euclidean distance between $\omega_{\mathbf{s}_1}$ and $\omega_{\mathbf{s}_2}$. The smaller the cooling profile distance is, the more similar the two cooling profiles are. With cooling profile distance we can do anomaly detection and k-means algorithm. Using standard anomaly detection and k-means techniques we can identify cooling profiles that do not conform to all expected others. With the distance metric, we can compare all cooling profiles pair-wise and similar distance metric clustering to one group in Fig. 5 & Fig. 6 and find outlier cooling profiles that are different from the others. We notify the system operators about these abnormal servers.

4 PRELIMINARY EVALUATION

4.1 Experiment setup

We performed evaluation in three different data centers with three different types of machines.

DC-A hosts about two hundred commodity 2U rack servers running Openstack based cloud computing environment. We collected data from 180 servers in different racks. It contains four rows of racks, six per row. Each rack only hosts 7-10 servers with other network switches, using a capacity about 4 KW per rack. The room has two air conditioner (AC) units which uses under-floor chilled air cooling.

DC-B is a similar sized machine room. It hosts four racks of Open Compute Project (OCP) [15] servers. We used one of the racks for our experiment. The rack contains 42 servers. Each rack has a total power of about 10KW. The room has a single air conditioner that chills the room using overhead cooling.

DC-C hosts over a hundred thousand servers serving real production jobs for a large-scale Internet service company. The building has an area of over a hundred thousand square meters, and has a total power capacity of 6 ~ 8 MW. We do not have information on the actual floor plans or cooling designs from the data center.

4.2 Accuracy of cooling profiles and false alarm assessment

As we mentioned, the cooling model is non-linear. To confirm the hypothesis, we compare our Gaussian Process Regression based cooling profile with three other algorithms. We collected raw data from 90+ servers in DC-A over 4 days, and used the data for training and testing. We applied 10-fold cross validation method, Table 1 shows the average accuracy and runtime of 4 algorithm in comparison. , we consider Gaussian process regression is a good trade off between accuracy and running-time in our cooling profile model.

As for false alarm assessment part, we implement cooling profile as an online failures detection and feed current data every 3 second, task uses around 3% CPU utilization. Each model monitoring their server for 24 hours and shows about 2.8% prediction error. Some errors come from abrupt drop or raise in CPU utilization, Others are from missing collected data when CPU utilization is extremely high. As we will show in next experiment, the model detects transient failures in 10 second, so we consider three continuous prediction error as alarm, and the average alarms are 2.1 per day.

Algorithms	Accuracy	Time
Linear Regression	90.12%	5 Sec
Support Vector Machines	79.84%	2 Min
Gaussian Process Regression	95.24%	35 Min
Conditional Random Field	94.64%	13 Hours

Table 1: Algorithms comparison

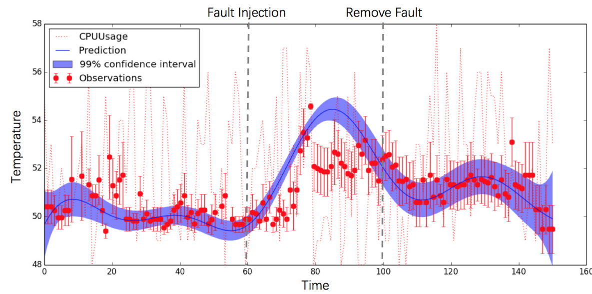

Figure 3: Transient cooling failure on our own servers. We seal the inlet and the outlet of one server at around 150s.

4.3 Detecting transient failures

In this section, we present our results detecting transient cooling failures. We first conducted a few fault injection experiments as micro benchmarks on how fast we can detect such a transient failure. Then we present a couple of real cooling failures we discovered in DC-A.

Fault injection experiments.Figure 3 summarizes that fault injection experiment results. At around the 60-th second we seal the inlet and the outlet and the our cooling profile rate goes above the realtime CPU temperature, At around the 100-th second we release the block of the inlet and the outlet and the the realtime CPU temperature gets back to the range of the 99% confidence. We want to emphasis that it is abnormal not because the prediction is higher but because the observed values are outside of the 99% confidence internal. The latency is within 10 seconds.

This is because in normal conditions, cooling profile did not collect data we seal vent before. Our algorithm puts the inlet/outlet temperature coefficients a very high position and anomaly CPU temperature raise the fan speed so the actual temperature lower than the prediction.

4.4 Detecting lasting problems

We build cooling profiles for the 42 servers in DC-B shown in figure 4 and 5, according to the cooling profiles from each server , figure 4 shows 42 servers‘ CPU temperature median value under cooling profile prediction and these servers contain in one rack, with the two obvious inflexions we determine K=3 when we use k-means clustering algorithm. Basically all the racks in DC-A and DC-B have been clustered to three parts depend on the inflexions. figure 5 shows K-means algorithm cluster all the servers into red yellow blue three groups and the black lines are the Euclidean distance between server to server. Each group has a unique temperature range and similar cooling profile but the red spot (marked

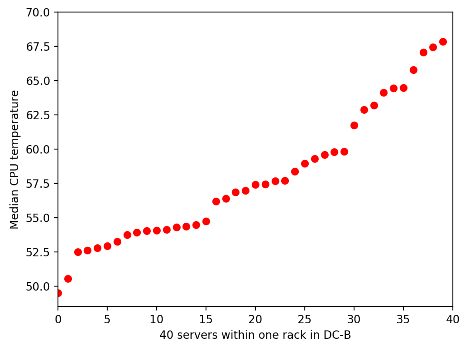

Figure 4: Servers' CPU temperature median value under cooling profile prediction shows two inflexions.

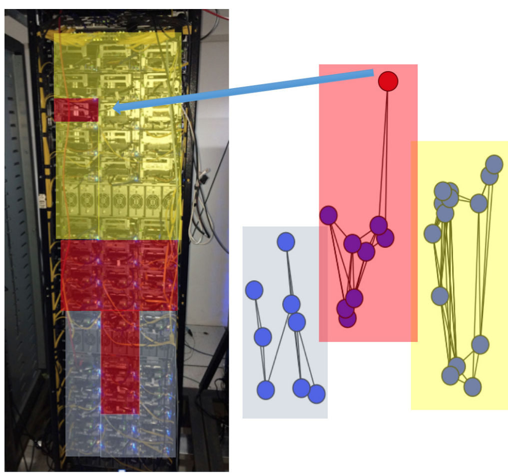

Figure 5: The anomaly detection cluster 42 servers in DC-B

with the arrow) are not similar with their neighbors matches the position of the rack. We checked the server and discovered that this server was the only one without the "shroud cover". After we place a wind hood into it, the cooling profile is no longer abnormal. In this case we didn't use the anomaly detection because with only one rack small sample, the server with missing "shroud cover" is similar to the poor cooling profile servers (the red groups) so the anomaly detection didn't get any outlier.

Fig. 6 shows the identified abnormal cooling profiles (red spots) from the normal ones (blue spots) in DC-C. Our anomaly detection algorithm identified 38 outliers, but the reason of these servers under poor profiles are not clear. In order to identify more information of these 38 outliers, with the same reason as last section we mentioned, k-means algorithm automatically cluster the red spots into two (several) groups. The first cooling profile plotted in figure 6(A) gives the normal cooling profile. From figure 6(B) we can see that even with low CPU utilization, the server CPU temperature still reaches 46°C, which means that this group of servers have poor cooling profiles at the very beginning. After informing the data center administrator, they took a closer inspection

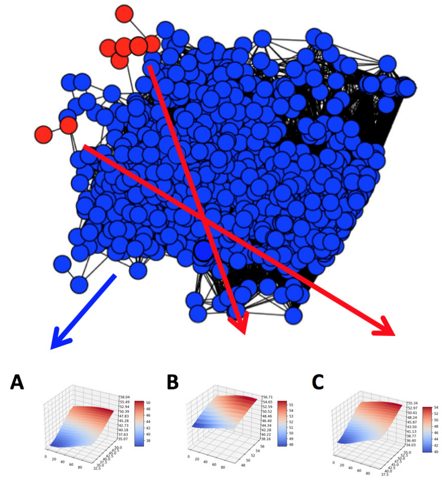

Figure 6: We used anomaly detection algorithm calculated over 1600 servers and found out two different kinds of design failures.

and revealed that their new design power supply modules are too close to these servers, which leads to thermal issues. Figure 6(C) shows an extreme case where the temperature gets unstable at moderate CPU utilization but with the low CPU utilization the temperature becomes normal. We suspect these server hardware have quality problems. It need to be noted that his situation does not mean the server cannot operate normally, but will affect the overall computing capacity.

5 CONCLUSIONS

Hidden cooling issues can still cause potential performance, energy efficiency or dependability problems and hard to detect. With our workload independent cooling profile model, we can capture the intrinsic cooling capability of each individual server. We can not only use the cooling profile to detect transient failure and also detect servers with lasting cooling problems. We eliminate the need for system identification by taking data-driven approach and only use readily available metrics while the data center is running production workload. We validate the general applicability of our approach using three data centers with vastly different scale, workload and server types.

As future work, we will expand the cooling profile approach to other non-server equipment, such as network devices whose workload is hard to measure. We would also like to use the server cooling profiles to improve job scheduling in the data center, minimizing the amount of cooling required to conserve energy. Last but not least, we want to build a data center planning tool so that people can place better heat-tolerant or lower power devices to the places with relatively poor thermal conditions.

REFERENCES

[1] Gray, J. (1986, January). Why do computers stop and what can be done about it?. In Symposium on reliability in distributed software and database systems (pp. 3-12).

[2] Patterson D, Brown A, Broadwell P, et al. Recovery-oriented computing (ROC): Motivation, definition, techniques, and case studies[R]. Technical Report UCB//CSD-02-1175, UC Berkeley Computer Science, 2002.

[3] Sankar S, Shaw M, Vaid K. Impact of temperature on hard disk drive reliability in large datacenters[J]. 2011:530-537.

[4] C. Belady, A. Rawson, J. Pfleuger, and T. Cader. The Green Grid Data Center Power Efficiency Metrics: PUE & DCiE. Technical report, Green Grid, 2008.

[5] Lawrence Berkeley National Labs. Benchmarking Data Centers. http://hightech.lbl.gov/ benchmarking-dc.html, December 2007.

[6] Intel I. and IA-32 Architectures Software Developer?s Manual[J]. Volume 3A: System Programming Guide, Part, 64, 1.

[7] N. El-Sayed, I. Stefanovici, G. Amvrosiadis, A. A. Hwang and B. Schroeder, "Temperature Management in Data Centers: Why Some (Might) Like It Hot," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 1, pp. 163-174, Jun. 2012.

[8] W. A. Abdelmaksoud, H. E. Khalifa, T.Q. Dang, R. R. Schmidt and M. Iyengar, "Improved CFD modeling of a small data center test cell," *12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, Jun. 2010.

[9] Frachtenberg E, Lee D, Magarelli M, et al. Thermal design in the open compute datacenter[C]//Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), 2012 13th IEEE Intersociety Conference on. IEEE, 2012: 530-538.

[10] H. Bhagwat, A. Singh and A. Sivasubramaniam, "Thermal influence indices: Causality metrics for efficient exploration of data center cooling," *2012 International Green Computing Conference (IGCC)*, Jun. 2012.

[11] Seymour M, Ikemoto S. Design and management of data center effectiveness, risks and costs[C]//Semiconductor Thermal Measurement and Management Symposium (SEMI- THERM), 2012 28th Annual IEEE. IEEE, 2012: 64-68.

[12] C. Kyosung, R. M. Galante, M. Ohadi and D. Cooper,"Measured and Simulated Energy Consumption Analysis of a Data Center on an Academic Campus," *29th Annual IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI- THERM)*, Mar. 2013.

[13] Chen J, Tan R, Wang Y, et al. "A high-fidelity temperature distribution forecasting system for data centers"*Real-Time Systems Symposium (RTSS)*, 2012 IEEE 33rd. IEEE, 2012: 215-224.

[14] Xiaodong Wang, Xiaorui Wang, Guoliang Xing, Cheng-Xian Lin "Leveraging Thermal Dynamics in Sensor Placement for Overheating Server Component Detection", *3rd IEEE International Green Computing Conference IGCC*, San Jose, USA, 2012, pp. 1-10

[15] Mulay, V."Open Compute Project: Server and Data Center Design." *Panel presentation at ASME InterPack*, San Jose, (2011).

[16] Anderson D, Dykes J, Riedel E. "More Than an Interface-SCSI vs. ATA"*FAST*. 2003, 2(00): 3.

[17] Sankar S, Shaw M, Vaid K." Impact of temperature on hard disk drive reliability in large datacenters", *Dependable Systems and Networks (DSN)*, 2011 IEEE/IFIP 41st International Conference on. IEEE, 2011: 530-537.

[18] Schroeder B, Pinheiro E, Weber W D. DRAM errors in the wild: a large-scale field study[C]// ACM SIGMETRICS Performance Evaluation Review. ACM, 2009, 37(1): 193-204.

[19] E. Kursun and C.-Y. Cher,"Temperature variation characterization and thermal management of multicore architectures", *IEEE/ACM MICRO*, 2009.

[20] M. Pedram and S. Nazarian,"Thermal modeling, analysis, and management in vlsi circuits: Principles and methods" *Proceedings of the IEEE*, vol. 94, no. 8, 2006.

[21] X. Fan, W. D. Weber and L. A. Barroso, "Power Provisioning for a Warehouse-sized Computer," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 13-23, Jun. 2007.

[22] H. Xu, C. Feng and B. Li, "Temperature aware workload management in geo-distributed datacenters," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 1, pp. 373-374, Jun. 2013.

[23] Moore J D, Chase J S, Ranganathan P, et al. "Making Scheduling" Cool": Temperature-Aware Workload Placement in Data Centers" *USENIX annual technical conference*, General Track. 2005: 61-75.

[24] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang and X. Zhu, "No power struggles: Coordinated multi-level power management for the data center," *ACM SIGARCH Computer Architecture News*, vol. 36, no. 1, pp. 48-59, Mar. 2008.

[25] D. Wang, C. Ren, S. Govindan, A. Sivasubramaniam, B. Urgaonkar, A. Kansal and K. Vaid, "ACE: Abstracting, characterizing and exploiting datacenter power demands," *IEEE International Symposium on Workload Characterization (IISWC)*, pp. 44-55, Sep. 2013.

[26] Callou G, Maciel P, Tutsch D, et al. "Models for dependability and sustainability analysis of data center cooling architectures", *Dependable Systems and Networks Workshops (DSN-W)*, 2012 IEEE/IFIP 42nd International Conference on. IEEE, 2012: 1-6.

[27] Beloglazov A, Buyya R. "Energy efficient resource management in virtualized cloud data centers", *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. IEEE Computer Society* , 2010: 826-831.

[28] Wang L, Von Laszewski G, Dayal J, et al. "Towards thermal aware workload scheduling in a data center", *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on. IEEE*, 2009: 116-122.

[29] Wang L, Khan S U, Dayal J." Thermal aware workload placement with task-temperature profiles in a data center", *The Journal of Supercomputing*, 2012, 61(3): 780-803.

[30] J. Gao and R. Jamidar, "Machine Learning Applications for Data Center Optimization," *Google White Paper*, Oct. 2014.

[31] J. Moore, J. S. Chase and P. Ranganathan, "Weatherman: Automated, online and predictive thermal mapping and management for data centers," *IEEE International Conference on Autonomic Computing (ICAC)*, pp. 155-164, Jun. 2006.

[32] Brunschwiler T, Smith B, Ruetsche E, et al. Toward zero-emission data centers through direct reuse of thermal energy[J]. IBM Journal of Research and Development, 2009, 53(3): 11: 1-11: 13.

[33] M. Arlitt, A. Palo and C. Bash, "Towards the Design and Operation of Net-Zero Energy Data Centers," *13th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*, Jun. 2012 .

[34] R. Zhou, Z. Wang, C. E. Bash, A. McReynolds, C. Hoover, R. Shih, N. Kumari and R. K. Sharma, "A Holistic and Optimal Approach for Data Center Cooling Management," *American Control Conference (ACC)*, Jul. 2011.

[35] G. Hoefel and C. Elkan, "Learning a Two-Stage SVM/CRF Sequence Classifier," *ACM 17th Conference on Information and Knowledge Management*, pp. 271-278, Oct. 2008.

[36] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey[J]. ACM computing surveys (CSUR), 2009, 41(3): 15.

[37] Schwenkler T. *Intelligent Platform Management Interface[J]. Sicheres Netzwerkmanagement: Konzepte, Protokolle, Tools*, 2006: 169-207.

[38] E. Upton and G. Halfacree, "Raspberry Pi user guide," *John Wiley & Sons*, Dec. 2013.

[39] *Hadoop A. Hadoop[J]*. 2009-03-06. http://hadoop. apache. org, 2011.

[40] G. Cole, "Estimating drive reliability in desktop computers and consumer electronics systems," *Seagate Technology Paper TP*, 338, 2000.

[41] B. Schroeder, E. Pinheiro and W. D. Weber, "DRAM Errors in the Wild: A Large-Scale Field Study," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, pp. 193-204, Jun. 2009.

[42] Phan L, Bhusal S, Lin C X. Improving Cooling Efficiency by Using Mixed Tiles to Control Airflow Uniformity of Perforated Tiles in a Data Center Model[C]//ASME 2017 Heat Transfer Summer Conference. American Society of Mechanical Engineers, 2017: V001T08A005-V001T08A005.

[43] Cho J, Yang J, Park W. Evaluation of air distribution system's airflow performance for cooling energy savings in high-density data centers[J]. Energy and buildings, 2014, 68: 270-279.

[44] Mok S, Joshi Y K, Kumar S, et al. Energy Simulations of Data Centers With Hybrid Liquid/Air Cooling and Waste Heat Re-Use[C]//ASME 2016 International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers, 2016: V008T10A008-V008T10A008.