Fine-grained Warm Water Cooling for Improving Datacenter Economy

Weixiang Jiang, Ziyang Jia, Sirui Feng, Fangming Liu*, and Hai Jin
National Engineering Research Center for Big Data Technology and System,
Key Laboratory of Services Computing Technology and System, Ministry of Education,
School of Computer Science and Technology, Huazhong University of Science and Technology, China

ABSTRACT

Driven by the increasing power consumption of datacenters, the industry is focusing more on water cooling for improving the energy efficiency. Using warm water to cool servers has been considered as an efficient method to reduce the cooling energy. However, warm water cooling may lead to the risk of cooling failure and its energy efficiency suffers from the thermal imbalance among servers, due to the lack of fine-grained cooling control. In this paper, we propose a hybrid cooling architecture design that incorporates thermoelectric cooler into the water cooling system, to deal with cooling mismatching in a fine-grained manner. We exploit the warm water cooling strategy and design an adaptive cooling control framework according to workload variations, to make water cooling system more economical for datacenters. We evaluate the hybrid water cooling design based on a real hardware prototype and cluster traces from Google and Alibaba. Compared with conventional water cooling system, our hybrid water cooling system can reduce the energy consumption by 58.72%~78.43% to handle the cooling mismatching.

CCS CONCEPTS

• **Hardware** → **Thermal issues**; **Enterprise level and data centers power issues**; • **Computer systems organization** → *Architectures.*

KEYWORDS

datacenter energy, water cooling, temperature, thermoelectric cooler

ACM Reference Format:
Weixiang Jiang, Ziyang Jia, Sirui Feng, Fangming Liu*, and Hai Jin. 2019. Fine-grained Warm Water Cooling for Improving Datacenter Economy. In *The 46th Annual International Symposium on Computer Architecture (ISCA '19), June 22–26, 2019, PHOENIX, AZ, USA.* ACM, 13 pages. https://doi.org/10.1145/3307650.3322236

*Corresponding author: Fangming Liu (fmliu@hust.edu.cn). This work was supported in part by the National Key Research & Development (R&D) Plan under grant 2017YFB1001703, in part by NSFC under Grant 61722206 and 61761136014 (and 392046569 of NSFC-DFG) and 61520106005, in part by the Fundamental Research Funds for the Central Universities under Grant 2017KFKJXX009 and 3004210116, and in part by the National Program for Support of Top-notch Young Professionals in National Program for Special Support of Eminent Professionals.

ISCA '19, June 22–26, 2019, PHOENIX, AZ, USA

ACM ISBN 978-1-4503-6669-4/19/06…$15.00
https://doi.org/10.1145/3307650.3322236

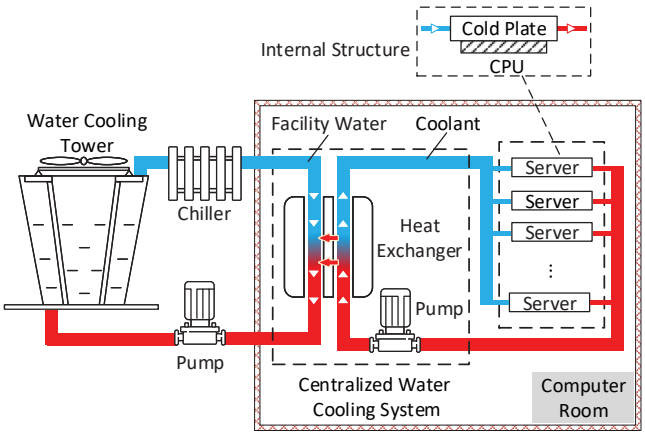


Figure 1: Water cooling architecture in a datacenter.

1 INTRODUCTION

Water cooling as a promising cooling technology has become the primary option for many datacenters to reduce the cooling cost [6, 17]. Compared with air cooling, water cooling carries high-density server implementation and better cooling efficiency. Fig. 1 shows a typical water cooling system in a datacenter. It mainly consists of two liquid loops: (*i*) the inner loop with coolant inside brings the heat from servers to the heat exchanger, and (*ii*) the outer loop with facility water inside transfers heat from the heat exchanger to the water cooling tower outdoors. The tower cools the water down via evaporation. Usually, datacenter uses cold water (e.g., 7℃~10℃) to cool servers [21]. After the tower removes a proportion of heat, the chiller goes on removing the remaining heat from the water. Such a water cooling system can serve hundreds even thousands of servers, and it could bring ~22% cooling energy reduction compared with air cooling by average[18].

State of the art water cooling systems cannot reduce the cooling energy to zero, because heat removed by the cooling tower is quite limited, and it requires chiller (which consumes an amount of energy) to help further cool the facility water, especially in hot days. Hence, raising the temperature of the facility water by running CPU at a higher temperature level can significantly reduce the energy consumption of chiller. It is reported that raising the temperature of facility water from 7℃~10℃ up to 18℃~20℃ can result in a cooling energy saving of ~40% [21]. Due to the low utilization of current datacenters [35, 38, 48], a more aggressive strategy, warm water (e.g., 40~45℃) cooling is suggested to reduce the cooling cost [14, 18]. CoolIT reports that warm water cooling

can deliver a rapid ROI (return on investment) within six months by average[15, 18]. Furthermore, ASHRAE Thermal Guidelines[7] "W5" suggests using >45℃ water to cool datacenters so that the outlet water is hot enough to heat buildings for datacenter heat recovery[13]. However, when some servers get high utilization in warm water cooling, they will exceed the safe operating temperature in a few seconds, while chiller needs a relatively long time (e.g., several minutes) to cool the water and delivers it to the overheated servers, which leads to cooling lag/mismatching and the risk of cooling failure. Though warm water cooling is attractive, its implementation is risky.

Due to the cost and technical difficulties, installing pumps in a centralized water cooling system for each server to enable fine-grained flow control is not practical, yet [16]. Another architecture which builds independent water circuits for each server to enable fine-grained flow control (e.g., IBM mainframe) is costly. Thus, the cooling control of the current water cooling system is centralized (i.e., controlling the temperature of the supplied water). It cannot provide a fine-grained cooling for each server separately, which brings the problem of hot spots in warm water cooling, i.e., some servers exceed the safe operating temperature while others don't. Hot spots hamper the energy efficiency of warm water cooling, because the chiller must reduce water temperature according to the highest temperature of the servers, while the other servers may not need such "cold" water.

Many software-based solutions have been considered to optimize the cooling efficiency in datacenters. *Workload deferral* can postpone some heat dissipation to shave the cooling peak. But this strategy cannot be applied to workloads with tight deadlines (e.g., interactive workloads), and it requires that a cooling peak must be followed by a cooling valley to provide redundant cooling capacity for the delayed workloads. *Workload throttling* (i.e., CPU frequency scaling) can alleviate the power usage of CPU as well as its cooling demand [30], but it can not be applied to workloads with performance guarantees. For example, the compute-optimized instance *c5* in Amazon EC2 promises to provide at least 3.0 GHz CPU frequency [9]. Servers have static power consumption even if they are idle [52]. So shutting down low-utilized servers by *workload consolidation* can eliminate the idle power consumption of servers and reduce the overall heat dissipation at the same time, but it requires lower water temperature as server's utilization becomes higher after consolidation, which increases the usage of chillers. Servers' power optimization does not necessarily bring cooling optimization [12]. Some online services such as web search require all servers to remain up regardless of traffic intensity [38]. Workload consolidation may not work for such interactive workloads, because the response time often suffers from higher utilization due to queuing effects [40]. *Workload balance* can also be considered to help alleviate the problem of hot spots, while it does not necessarily bring the thermal balance. Workload balance usually aims to improve the performance (e.g., minimizing the makespan of jobs). Its balance usually means processing speed balance rather than utilization balance (e.g., in a heterogeneous datacenter). As shown later by real-world cluster traces from 2011 to 2017 in Sec. 2, the utilization imbalance in a cluster is a common phenomenon [57].

Regardless of the above limitations, a thermal-aware workload management strategy may conflict with other optimization strategies (e.g., network-aware workload placement [5, 51]), leading to a trade-off between the performance and the cooling energy. The workloads in a datacenter are usually managed by software engineers who care more about performance, while cooling systems are managed by different departments or engineers, which makes thermal-aware workload management more difficult in practice. *Therefore, can we find a solution that addresses the challenges (i.e., the risk of warm water cooling and the hot spot problem) in warm water cooling, no matter how datacenters manage their workloads?*

To tackle the above challenges, we propose a hybrid water cooling system which integrates each CPU with a thermoelectric cooler (TEC), which provides extra cooling for individual servers dynamically. To the best of our knowledge, we are the first to explore optimizing warm water cooling in a datacenter. Specifically, we make the following contributions in this paper:

(1) We propose a hybrid water cooling system that incorporates TEC into the existing water cooling system and build a prototype to validate its availability. The TEC can provide extra cooling capacity for the water cooling system in a fine-grained manner, which allows operating warm water cooling strategy safely and efficiently.

(2) By exploring the spatial and temporal workload distribution in datacenters, we design an adaptive cooling control mechanism which can adjust the cooling provision strategy according to the variation of workloads. Specifically, our adaptive cooling control mechanism uses the chiller to provide a base cooling capacity and TEC to provide a dynamic cooling capacity to handle the rest cooling mismatching.

(3) We evaluate the proposed system based on a hardware prototype and server traces from Alibaba and Google. The results demonstrate that our hybrid water cooling system can reduce energy consumption by 58.72%~78.43% to handle the cooling mismatching, compared to the conventional water cooling system.

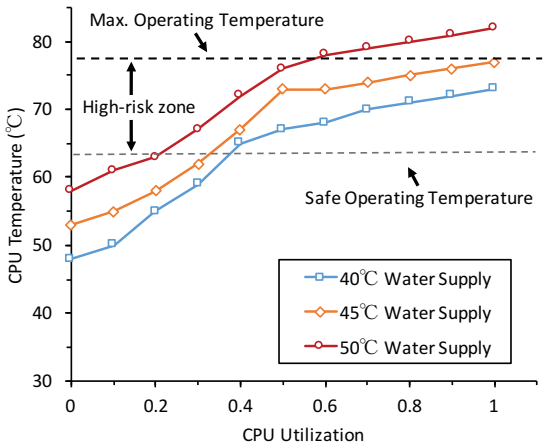
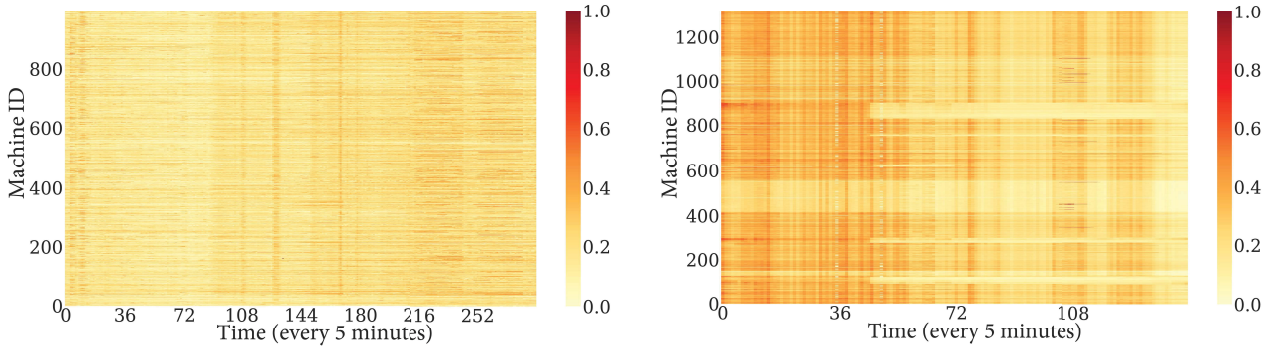


Figure 2: Temperature variation of CPU with different utilization and water temperature (Type: Xeon E5-2650 V3, Frequency Governor: powersave, Maximum Operating Temperature: 78.9℃ [56]).



(a) Utilization of Google cluster in 2011

(b) Utilization of Alibaba cluster in 2017

Figure 3: CPU utilization (from 0 to 1) analysis of cluster traces.

CPU Utilization	Temperature (℃)																			
	node1		node2		node3		node4		node5		node6		node7		node8		node9		node10	
	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1	CPU0	CPU1
0%	31	31	32	35	33	32	32	32	31	32	33	32	32	31	33	32	31	31	30	29
20%	35	35	35	38	34	33	37	36	33	34	36	35	34	34	35	35	32	31	32	30
40%	38	38	36	41	38	36	39	39	35	36	38	37	36	36	37	37	37	35	35	33
60%	39	39	39	44	40	37	42	41	37	38	40	39	38	38	39	38	38	37	37	35
80%	39	39	40	46	40	38	43	42	38	38	40	40	38	39	40	38	39	38	38	36
100%	40	40	41	48	42	40	44	44	38	39	42	42	40	40	41	40	41	39	40	38

Figure 4: Server nodes running at the same CPU utilization have different temperatures in a centralized water cooling system (supplied coolant temperature is 20℃).

2 BACKGROUND AND MOTIVATION

In this section, we introduce the detailed background of warm water cooling strategy and fine-grained cooling.

2.1 Warm water cooling

Except for a few datacenters, most datacenters are located in warm areas [45], and warm water cooling has attracted growing attentions [14, 18]. We conducted a measurement of CPU temperature variation with different utilization and water temperature as shown in Fig. 2. When CPU has a high utilization, its temperature is close to the maximum operating temperature. Note that the CPU temperature increases slower when utilization is larger than 50%, because CPU frequency governor "powersave" provided by Intel *p_state* will automatically scale the CPU frequency for energy saving, and it locks the CPU frequency at 2.3 GHz in our measurement when utilization is larger than 50%.

Pro-longed operation at close to the maximum operating temperature may cause system instability and shorten the CPU lifespan [55]. Usually, a safe operating temperature (e.g., 80% of the maximum operating temperature) will be set as the upper bound. Warm water may cause damages to a CPU when the CPU's utilization is high, but a datacenter usually has a low utilization [35] (e.g., 26.32% by average in Alibaba cluster [3, 39]). Hence, 40~45℃ and even 50℃ water is possible to cool servers to further reduce the usage of chillers. Though the average utilization is low, when we break down the server usage of both Google and Alibaba clusters as shown in Fig. 3, many servers with >40% utilization exceed the safe operating temperature (63℃) according to the results in Fig.2. Higher water temperature may lead to the risk of overheating for some servers.

2.2 Fine-grained cooling control in datacenter

Two factors lead to the thermal imbalance a datacenter. First, the server's utilization in a datacenter is spatial imbalanced as shown in Fig. 3. Google cluster trace in 2011 provides the utilization of 12.5k machines over one month [23]. As an illustration, we only show 1000 machines' utilization for 24 hours. Alibaba cluster trace in 2017 provides the utilization of 1313 machines during 12 hours [3, 39]. We can see that the CPU utilization in a datacenter is imbalanced in space, which causes the thermal imbalance. Second, the cooling efficiency of the current water cooling system is also spatial imbalanced. We conducted a real-world measurement on ten Sugon TC4600E-LP server nodes, whose CPUs are cooled by a centralized water cooling system. We stress the ten nodes with the same CPU load and record their CPU temperatures as shown in Fig. 4. We can see that some nodes are much hotter than others (e.g., when CPUs are stressed at 100% utilization, the temperature of CPU1 in node2

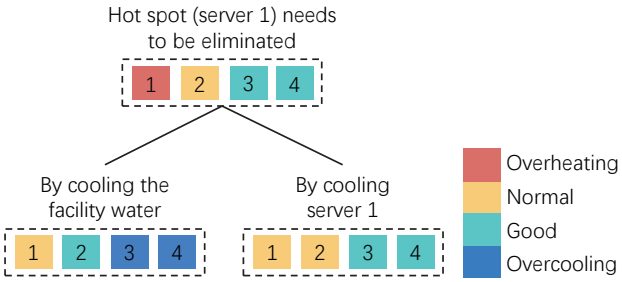


Figure 5: Solutions to eliminating hot spot in water cooling datacenter.

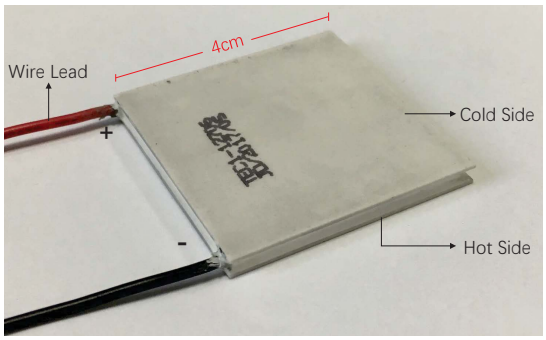


Figure 6: Thermoelectric cooler.

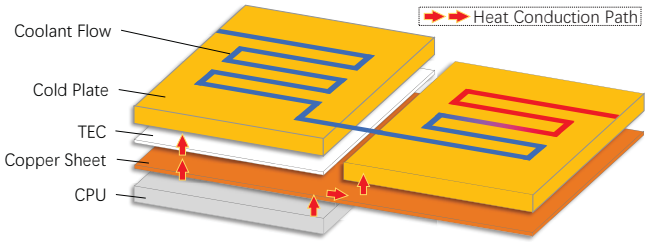


Figure 7: TEC integration inside a server.

is 10℃ higher than CPU0 in node5). This is because hydraulic pressure varies from place to place, due to the different distances to the centralized pump, and servers share the main coolant tube, which results in non-uniform flow rates inside each server. The above two factors unavoidably make some servers become hot spots in space.

It should be noted that the thermal imbalance is not a problem in a conventional water cooling system, because even though we fully stress each CPU, its temperature is still quite low due to the cold water as shown in Fig. 4. But in warm water cooling, the thermal imbalance will incur hot spots [1], which hamper the cooling efficiency. Fig. 5 shows an example. If the current water cooling system adopts warm water cooling and server 1 is overheated, the only method is to reduce the temperature of the facility water, which overcools server 3 and server 4. A more energy-efficient way is to only cool server 1. Thus, a fine-grained and precise cooling control is demanded.

In the current water cooling system, implementing distributed pumps to adjust the flow rate inside each server is a possible solution to provide fine-grained cooling control, but it is technically challenging. For example, servers share the same coolant tubes, and raising the flow rate in one server will cause insufficient coolant flow in neighboring servers. Distributed pump will cause a series of liquid pressure problems and is costly, which is not suggested by the manufacturer [16]. So can we find a solution to enable fine-grained cooling based on the current water cooling system?

3 HYBRID WATER COOLING: CHARACTERIZATION AND KEY DESIGN

In this section, we introduce a hybrid water cooling design integrated with TEC to enable fine-grained cooling.

3.1 What is TEC?

A thermoelectric cooler (TEC) is a semiconductor-based electronic component, which operates by the Peltier effect, as shown in Fig. 6. When applying a direct current to a TEC, the heat will be moved from one side to the other, generating two opposite faces: cold and hot sides. The cold side can be used for cooling.

Physical characteristic: a TEC is powered by direct current below 12 V, and it can provide different cooling capacity by adjusting its power supply (changing the supplied voltage from 0~12 V). Its power requirement is the same as conventional CPU fan, so a TEC can be connected to the 4-pin power connector on the motherboard to get a controllable power supply. Moreover, its size is $4 \times 4\ cm$, which is the same as the size of CPU. Thus, it can be attached to the CPU directly.

Cost analysis: The price of TEC is quite low. For example, the TEC (model: TEC1-12706) in Fig. 6 is ~$2 bought from *Alibaba.com*[4]. The Mean Time Before Failure (MTBF) of TEC of modern manufactures is usually not less than $250k \sim 300k$ hours [41], which is much larger than the lifespan of a server. So TEC can act as a cheap and reliable cooler in a computing system.

Limitation: TEC consumes energy, and its peak power consumption is about 60 W. So it should be carefully used for energy saving. Besides, TEC requires removing heat from the hot side to prevent TEC from overheating damage. Besides, TEC has a poor thermal conductivity when it is turned off. To enable a flexible switch between TEC cooling and water cooling, we need a new hardware design.

3.2 Integrating TEC into server

Fig. 7 shows our hybrid water cooling design. The original water cooling only contains a cold plate, attached to CPU directly. Our hybrid water cooling system has three more components: copper sheet, TEC, and another cold plate. The heat dissipated by CPU is delivered to the copper sheet. When the CPU temperature is smaller than safe operating temperature, the TEC is turned off and does not consume any energy. Then the heat is transferred from radiator copper to the cold plate (i.e., the right heat conduction path in Fig. 7). Note that TEC hardly conducts heat when it is turned off.

[1] A hot spot is a server whose CPU temperature exceeds a pre-set safe value.

When CPU has high utilization, and the coolant is too hot to cool the CPU, TEC starts to work and transfer the heat to the other cold plate (i.e., the left heat conduction path in Fig. 7). In this way, we can seamlessly switch the cooling method between TEC and water cooling.

Placing TEC directly in contact with a copper sheet to enable the switch between water cooling and TEC cooling has a drawback: the heat in the right cold plate may return to the copper sheet, as the copper sheet becomes much colder when TEC works. But as our measurement on a prototype (shown in Sec. 3.4 later), TEC still achieves a high cooling rate.

3.3 TEC VS. PCM

Previous work [49] has explored using phase change materials (PCM) to shave the peak cooling demand in the air cooling system. PCM absorbs/releases large amounts of heat by melting/solidifying (i.e., phase changing) at a certain temperature (i.e., melting temperature). So PCM only works when the ambient temperature is higher/lower than its melting temperature. Note that water cooling can absorb a lot of heat without phase change due to water's excellent heat capacity, which is different from the working principle of PCM.

There are three basic elements that need to be considered when implementing PCM to warm water cooling: (*i*) the melting temperature, which determines when PCM works; (*ii*) the speed of heat absorbing, which guarantees the CPU will not exceed its safe operating temperature; (*iii*) capacity of heat absorbing, which determines how long PCM can work.

By comparison with other various PCMs, commercial grade paraffin is chosen as the best material due to its high heat of fusion and suitable melting temperature [49]. For element (*i*), warm water cooling keeps CPU close to its safe operating temperature (e.g., ~60℃). Hence, PCM's melting temperature should be approximately the same as it. For element (*ii*), the thermal conductivity of paraffin is about 0.25 W/m℃ which is 2~3 times lower than water's (about 0.6 W/m℃) [53, 54]. According to the measurement in Fig. 4, 20℃ water can cool CPU to 38~48℃ when CPU is fully stressed. Theoretically, PCM's melting temperature must be much lower than 20℃ to achieve the same cooling effect, which is against element (*i*). To further validate the above analysis, we bought commercial grade paraffin from *Alibaba.com* with melting temperature of 58℃ and attached it to CPU directly. When CPU utilization was 0%, the PCM melted relatively slow, and the temperature of CPU grew by the speed of ~5℃/minute, and finally exceeded the maximum operating temperature. When we stressed the CPU utilization up to 80%, the PCM melted into liquid much faster, but the CPU's temperature became higher than 85℃ in 5 seconds and still kept rising. Thus, paraffin cannot handle the CPU's temperature due to its poor thermal conductivity. Note that we provided a sufficient amount of paraffin, and the solid paraffin was always pressed tightly on the CPU during the test. Besides, it is difficult to find a stable and cheap solid material with a low melting temperature and a thermal conductivity comparable to that of water in reality [49, 53]. For element (*iii*), suppose absorbing 1/5 heat (i.e., 20 W) dissipated by a modern CPU with thermal design power (about 100 W [56]) for just 15 minutes, it requires about $4cm \times 3cm \times 10cm$ paraffin according to its heat of fusion (200 J/g) and density (0.7 to 0.8 g/ml) [49]. Even if paraffin can satisfy element (*ii*), it requires too much space in a server not to mention a longer working time. This is against the goal of making sever smaller to enable high-density implementation in a water-cooled datacenter.

Though a further work [50] proposes to control the melting of PCM by workload placement to change the temperature inside a server, PCM is still not suitable to handle CPU's temperature in warm water cooling due to the above elements, and it affects the workload management strategies.

In warm water cooling, the inlet water is not "cold" enough to cool the CPU. Thus, it requires an extra cooling solution, which can keep up with the heat generating rate of a CPU. In the PCM-based work, the inlet air is always cold enough to cool CPU because of the use of air cooling system. PCM is placed at the air outlet and absorbs a part of the heat to reduce the peak load of the air cooling system so that datacenter can reduce the capital cost of the air cooling system. But it cannot handle the CPU temperature due to its poor thermal conductivity. Maybe PCM can be placed outside servers (e.g., cooling tower) to absorb heat from the water and reduce the overhead of the cooling tower, especially in hot days.

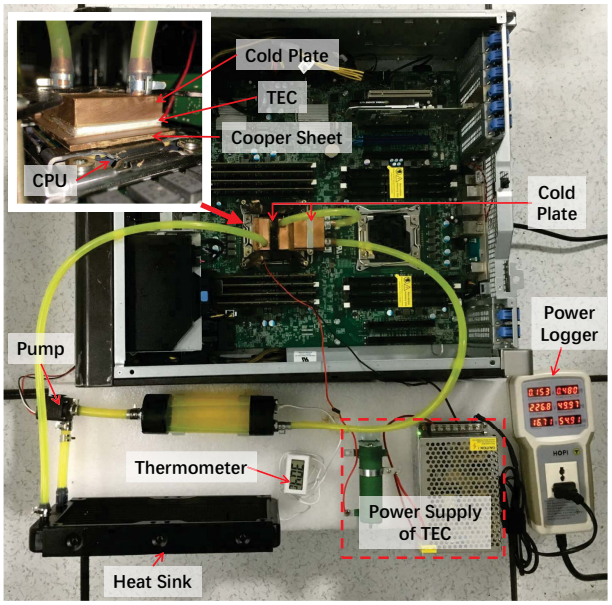


Figure 8: Prototype of a hybrid water cooling system.

3.4 Modeling the cooling capacity of hybrid water cooling

Two critical properties of TEC in this system need to be carefully investigated: *power consumption* and *cooling capacity*.

We build up a hybrid water cooling test-bed in a Dell server as shown in Fig. 8. The server is equipped with an Intel Xeon E5 2650 V3 CPU. We use a default set of CPU power governor:"powersave". The hardware attached to CPU shown in the upper left corner conforms to the design in Fig. 7: copper sheet, TEC, and two cold plates.

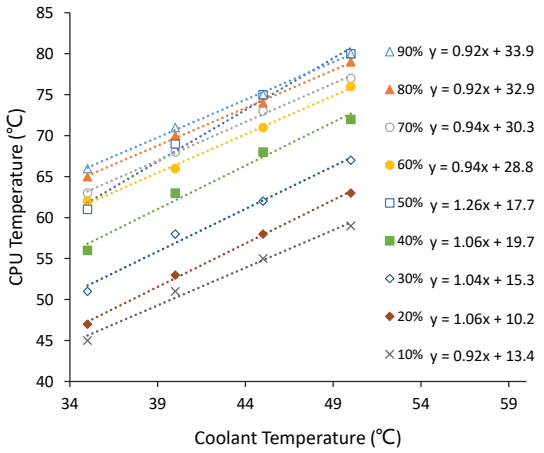


Figure 9: CPU temperature grows linearly with coolant temperature at different CPU usage (TEC is turned off).

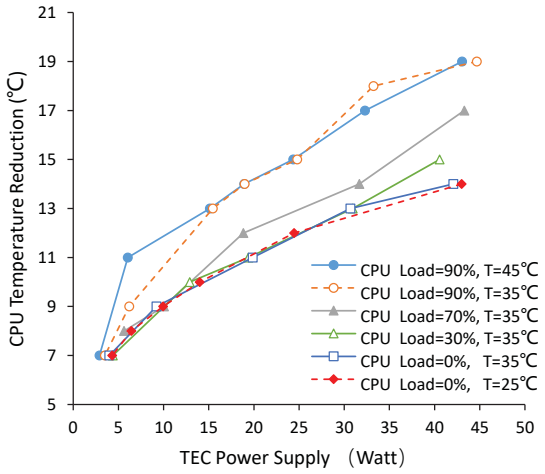


Figure 10: Cooling efficiency of TEC (T is the coolant temperature).

In the lower left corner is a water circulation system, consisting of a pump, a heat sink (acting as a cooling tower), and a thermometer to monitor the temperature of the coolant. Especially, we build an external power supply for TEC as shown in the lower right corner in Fig. 8 so that we can use a power logger to record TEC's power consumption. It should be noted that all the hardware components except the TEC, copper sheet and cold plates, are not part of our hybrid water cooling design, and they are built for evaluating the performance of TEC.

We first test the water cooling efficiency in the case that TEC is turned off. In the conventional design of a water cooling system, the cold plate is directly attached to CPU. But in our prototype, both cold plate and CPU are attached to a copper sheet. It should be noted that the copper sheet has little effect on the cooling efficiency according to our measurement, as the temperature of CPU is the same after removing the copper sheet and attaching the CPU to cold plate directly.

Fig. 9 shows the measurement results of CPU temperature variation with coolant temperature at different CPU usage. We stress the CPU at different utilization and record the CPU temperature at different coolant temperature supply. We observe that when the CPU utilization is fixed, the temperature of CPU grows linearly with coolant temperature. More interestingly, the slopes are approximately equal to 1 in most cases as shown in Fig. 9, which means if we want to reduce the CPU temperature by ΔT in a conventional water cooling system, then we need to reduce the coolant temperature by ΔT, too.

Then we measure the cooling efficiency of TEC in our hybrid design. We observe the cooling capacity of TEC grows with its power supply. Moreover, we explore the effects of CPU load and water temperature on the cooling capacity of TEC. Fig. 10 shows the CPU temperature reduced by TEC in different scenarios. We can see that the coolant temperature has little effect on the cooling capacity of TEC (by comparing CPU Load=90%, T=45℃ and T=35℃; CPU Load=0%, T=35℃ and T=25℃), while the CPU load has a significant effect on the cooling capacity of TEC. When given the same power supply, TEC can reduce more CPU temperature when CPU utilization is higher. Overall, TEC in our hybrid design is capable to cool the CPU by reducing the CPU temperature up to 19℃.

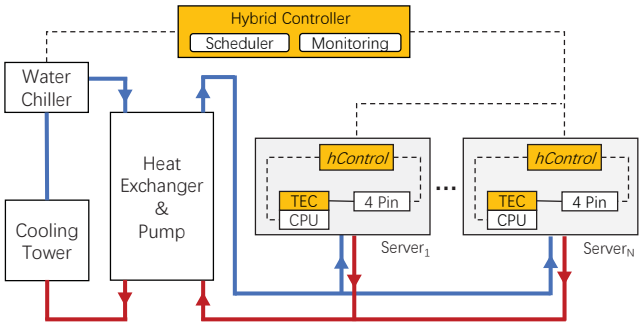


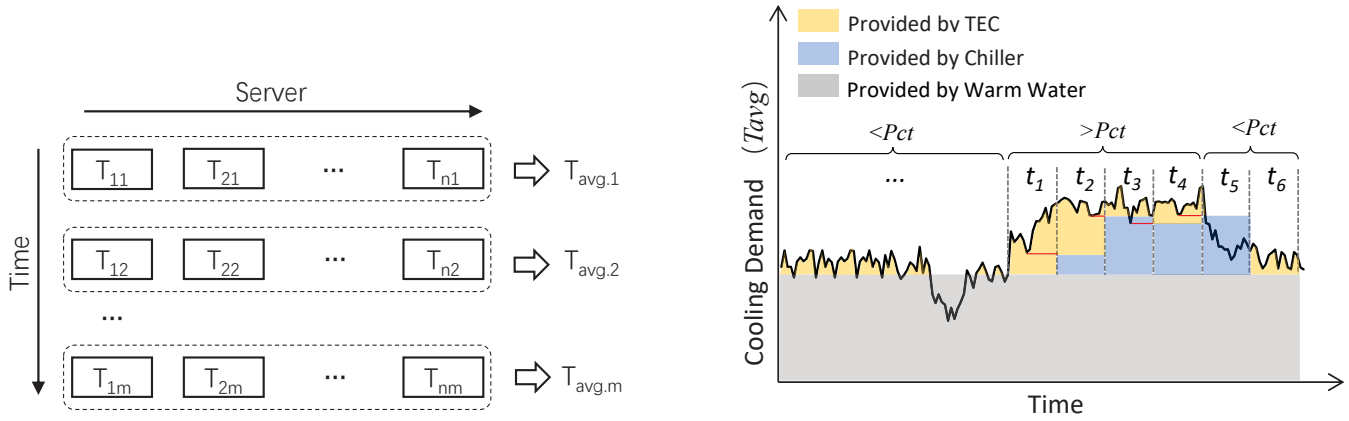
Figure 11: An overview of hybrid water cooling architecture.

4 SYSTEM ARCHITECTURE

In this section, we introduce how to implement the hybrid water cooling system in a datacenter and enable flexible cooling control.

Overview: As is shown in Fig. 11, it has three major parts in our hybrid water cooling system. The first part is the water chiller and cooling tower, which is the same as the conventional water cooling system. The second is the additional TEC modules which have been demonstrated in Sec. 3.2. It can realize real-time cooling in a fine-grained manner without lowering the temperature of water. The last part is the controllers, making proper cooling decisions according to the workload variation and optimizing the cooling energy efficiency.

hControl: it uses the `lm_sensors` tool in Linux to collect the temperatures of CPUs, and send them to the *Hybrid Controller*. As mentioned before, the power requirement of TEC is the same as traditional CPU fan's, so it can be connected to the 4-pin power connector on the motherboard. The 4-pin power connector uses a



(a) The temperature map of servers in a cooling control interval.

(b) Adaptive cooling control in the hybrid water cooling system.

Figure 12: Adaptive hybrid cooling control.

pulse-width modulation (PWM) input signal, which gives the ability to adjust its power output to control fan speed, and `lm_sensors` provides `pwmconfig` and `fancontrol` to manually configure 4-pin power output. By taking advantage of the above tools, *Power controller* can be easily implemented to adjust the cooling capacity of TEC.

Hybrid controller: *Monitoring* collects the CPU temperature information from *hControl*. *Scheduler* is the key decision-making component that controls water chiller and TEC according to the temperature information from *Monitoring*.

Obviously, the efficiency of the whole cooling system highly depends on the cooling strategy of *Scheduler*. Next, we introduce how to design the cooling strategy of the hybrid water cooling system.

5 COOLING MANAGEMENT

In this section, we present the cooling control for the hybrid water cooling system. Suppose the cooling tower can continuously provide warm water of temperature $T_{WarmWater}$ which can cool the CPU i to temperature T_{ij} at time instance j. Once a server's utilization increases suddenly and its temperature exceeds a pre-defined safe temperature T_{safe}, TEC can respond to the sudden cooling mismatching in real time, eliminating the cooling mismatching of conventional water cooling.

5.1 Adaptive hybrid cooling control

However, the above cooling control is not the most energy-efficient solution. As we can see from Fig. 10, there is a trend that the efficiency of TEC (i.e., temperature reduction/watt) drops when achieving more reduction in CPU temperature, and we find that using chiller and TEC to jointly handle the cooling mismatching is more energy-efficient. The question is: to achieve the best cooling energy efficiency, when should the chiller work and how much cooling capacity should the chiller provide?

If we have pre-knowledge of the incoming cooling demands, we can compare the cooling energy when the chiller provides different proportions of cooling capacity and figure out an optimal cooling strategy in advance. However, cooling demand prediction is not easy, especially in public datacenters where multiple users may submit their jobs irregularly.

To solve the above problem, we propose an adaptive cooling control mechanism. For each time instance j, we can get the temperature T_{ij} of CPU i[2]. Then we have a CPU temperature map as shown in Fig 12(a). We call T_{ij} that exceeds the pre-defined safe temperature as a hot spot, which needs to be cooled.

For the convenience of illustration, we use the average CPU temperature T_{avg} in Fig. 12(a) to measure the overall cooling demands of each time instance. We only activate the chiller when the percentage of hot spots is larger than a pre-defined threshold *Pct* (e.g., 80%). As shown in Fig. 12(b), when the percentage of hot spots is larger than *Pct*, the cooling mismatching can be handled as below:

- *Step 1.* in the first cooling control interval when the percentage of hot spots is larger than *Pct* (e.g., t_1), we only use the TEC to cool the server, because TEC can cool the CPU in real-time.
- *Step 2.* at the beginning of the next interval (e.g., t_2), if the percentage of hot spots in the previous cooling control interval (i.e., t_1) is larger than *Pct*, chiller provides the cooling capacity according to the lowest cooling demand in the previous interval. That is to set the chiller to reduce the water temperature by $\Delta T = min\{T_{avg,1}, T_{avg,2}..., T_{avg,m}\} - T_{safe}$. The rest cooling mismatching is handled by TEC; otherwise, go to *Step 3*.
- *Step 3.* If the percentage of hot spots in the previous interval is smaller than *Pct* (e.g., t_5), only TEC handles the cooling mismatching in this interval (e.g., t_6).

The above cooling control is based on the CPU temperatures in the previous cooling control interval. It can dynamically schedule the cooling capacity without pre-knowledge of workloads and achieve a smarter cooling control.

[2]CPU's temperature can be easily monitored by many system tools, e.g., `lm_sensors` in Linux.

6 EVALUATION METHODOLOGY

As we don't own a real-world datacenter with hybrid water cooling system, we conduct experiments based on the energy and thermal models obtained from the prototype and simulate the energy usage by using cluster traces from Alibaba and Google. It is important to note that in a real-world implementation, the following models are not needed. For example, CPU's temperature can be monitored by system tools and the energy consumption of chiller can be measured by power meter directly. For TEC energy model, we adopt its worst cooling case to guarantee the validity. Thus, our solution is easy to be evaluated in real-world datacenters.

6.1 Modeling CPU temperature

We consider a homogeneous datacenter where each server is equipped with Xeon-E5-2650 V3 CPU, and CPUs are configured with Intel frequency governor "powersave". Then according to our measurement in Fig.2, we can model the CPU temperature T_{CPU} and its usage u as a piecewise linear function:

$$T_{CPU}(u) = \begin{cases} 36.29u + 57.09, & \text{if } u \in [0, 0.5), T_{water} = 50°\text{C} \\ 10u + 72, & \text{if } u \in [0.5, 1], T_{water} = 50°\text{C} \end{cases} \quad (1)$$

We choose homogeneous datacenter because it is easy to model the CPUs' temperatures for evaluation. Our solution still works in a heterogeneous datacenter. As we mentioned, in a real-world implementation, CPU's temperature can be monitored directly by system tools, and our hybrid cooling system is managed based on the temperature of CPU. The difference is that a heterogeneous datacenter may generate a temperature map with different values. Hence, we use different workload traces to generate different temperature maps for evaluation later.

On the other hand, the temperature of CPU grows linearly with the temperature of coolant as shown in Fig. 9 and it can be expressed as:

$$T_{CPU} = T_{water} + D(u) \quad (2)$$

$D(u)$ is the intercept when CPU utilization is u, as shown in Fig. 9.

With the above two thermal models, we can simulate the cooling demands (i.e., each CPU's temperature) according to the CPU usage from cluster traces.

6.2 Modeling cooling energy consumption

Conventional water cooling: When some servers get computation-intensive jobs during time t, and their CPU temperature exceeds the pre-defined safe temperature. Suppose the highest CPU temperature exceeds the safe temperature by ΔT. To meet the cooling demand, the chiller needs to cool the water that flows into the heat exchanger and reduce water temperature by ΔT. Suppose the water flow speed is F, measured in m^3/h, then the energy consumption of chiller during time t can be denoted as:

$$E_{ConventionalWaterCooling} = C_{water} * \Delta T * F * t * \rho / COP_{chiller}. \quad (3)$$

$C_{water} = 4.2 \times 10^3$ J/(kg·℃) is the heat capacity of water. It means 4.2×10^3 Joule heat needs to be added to (or removed from) 1-kilogram water to change water temperature by 1 ℃. $F * t$ is the total water volume that flows into heat exchanger during time t and ρ is the density of water. $COP_{chiller}$ (coefficient of performance) is a concept of thermodynamics that describes the energy efficiency of a chiller. It is determined by the ratio between the heat that chiller removes from water and the energy it consumes (i.e., COP=Heat removed by chiller/Energy consumed by chiller).

TEC: When using the TEC to handle the same scenario, each server can be cooled separately. As our measurement shown in Sec. 3.4, TEC shows different cooling capacities under different CPU load. Given the same power supply, it can reduce more temperature when the CPU load is higher. In order not to lose generality, we adopt the worst case to model the cooling capacity of TEC, that is TEC's cooling capacity when CPU Load=0%, T=25℃ in Fig. 10. To better fit the cooling capacity and power characteristic of TEC, we use a piecewise quadratic function to model the curve:

$$P_{TEC}(\Delta T) = \begin{cases} 0.092\Delta T^2 + 0.034\Delta T, & \text{if } \Delta T \in [0, 8] \\ 0.89\Delta T^2 - 14.16\Delta T + 65.43, & \text{if } \Delta T \in (8, 14] \end{cases} \quad (4)$$

For each second, suppose N servers exceed the pre-defined safe temperature by $\Delta T_1, \Delta T_2, \ldots \Delta T_N$, respectively. Then the total power consumption of TEC in this second can be expressed as:

$$E_{TEC} = \sum_{i=1,\ldots,N} P_{TEC}(\Delta T_i) \quad (5)$$

6.3 Evaluation Setup

To better evaluate the performance of our hybrid water cooling system and the cooling control mechanism, we set three types of traces and three baselines in the evaluation.

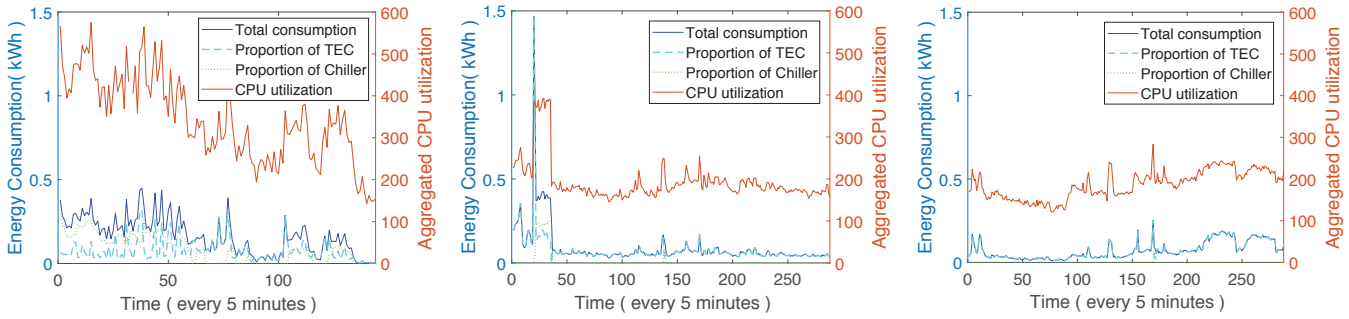
Drastic: this trace is from Alibaba cluster [39], which contains 1313 servers' CPU usage in 12 hours. Its overall CPU usage has drastic and frequent fluctuations.

Irregular: Google trace provides 12.5k servers' CPU usage during a month. We select 1000 servers' CPU usage for 24 hours to compose an abnormal workload trace. Its overall CPU usage is relatively common, but with occasional high peaks.

Common: similarly, we select another 1000 servers for 24 hours from Google trace to compose an overall cooling demand whose degree of fluctuation is smaller than *Drastic*. These three traces will be demonstrated later in the evaluation results.

Optimal baseline (hybridOpt): if we have pre-knowledge of the incoming cooling demand or implement some prediction algorithms, then we can find an optimal cooling control solution in advance which can minimize our hybrid cooling energy. As we already have the traces, by trying different settings of chiller (i.e., trying different water temperature reduction ΔT of chiller from 0), we can figure out the strategy with minimum cooling energy as the optimal baseline.

Chiller baseline (chiller): For the conventional water cooling system, it can only use the chiller to handle the cooling mismatching. Similarly, if we have pre-knowledge of the incoming cooling demand, we set the chiller according to the highest cooling demand



(a) Drastic (b) Irregular (c) Common

Figure 13: Energy usage pattern of hybrid water cooling system (T_{water} = 50°C, Pct = 80%, cooling control interval is 15 minutes).

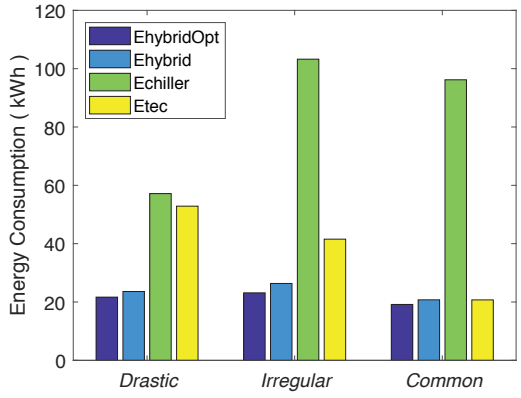


Figure 14: Energy consumption to handle cooling mismatching of different strategies (T_{water} = 50°C, Pct = 80%, cooling control interval is 15 minutes).

of the incoming cooling control interval and conduct its cooling energy consumption as another baseline.

TEC baseline (TEC): We can only use TEC to handle the cooling mismatching. This will result in another energy consumption as a new baseline.

We set up the other parameters based on the following considerations:

- COP of chiller ($COP_{chiller}$): Minimum COP requirements of different chillers in the US is 2.7~3.6[37]. Here, we assume COP of chiller is 3.6.
- Water flow speed (F): In water-cooled Sugon TC4600ELP server we mentioned in Sec. 2.2, its water flow speed is set as 0.5 m^3/h. We adopt the same water flow in our evaluation.
- Safe operating temperature of CPU (T_{safe}): Pro-longed operation at close to the maximum temperatures may cause CPU performance degradation and shorten the CPU lifespan [55]. To ensure the safety and performance, we define T_{safe} as 80% of CPU's maximum operating temperature. For Xeon-E5-2650 V3 CPU in our experiments, its safe operating temperature is 63℃ [56].

It should be noted that the models and parameters in our evaluation come from real-world system settings and measurements.

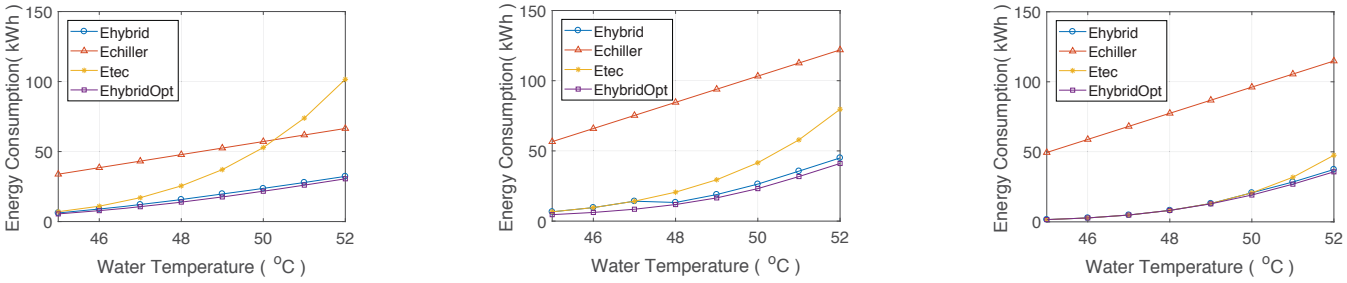
7 EVALUATION RESULTS

In this section, we analyze the energy consumption of our hybrid water cooling design and compare it with different baselines using different workload traces.

Energy pattern of hybrid water cooling system: Fig. 13 shows the workload traces (*drastic*, *irregular*, *common*) and the corresponding energy consumption of our hybrid water cooling system under three different workloads. We can see the total energy consumption follows the variation of the aggregated CPU utilization. It avoids providing unnecessary cooling capacity and energy waste. It should be noted that the aggregated CPU utilization can not fully describe the cooling demand of the CPUs, as it misses information of local hot spots. Especially, the energy proportion of chiller in Fig. 13(a) is higher than that in Fig. 13(b) and Fig. 13(c). This is because its overall CPU usage is much higher, resulting in more opportunities to trigger chiller. In Fig. 13(b) and Fig. 13(c), we can see the chiller's energy consumption maintains 0 at most time, and it is only triggered when high peak comes. This is because its overall CPU utilization is relatively low (less than 200) at most of the time and the percentage of hot spots is less than our setting 80% in these periods. Thus, the chiller is not activated.

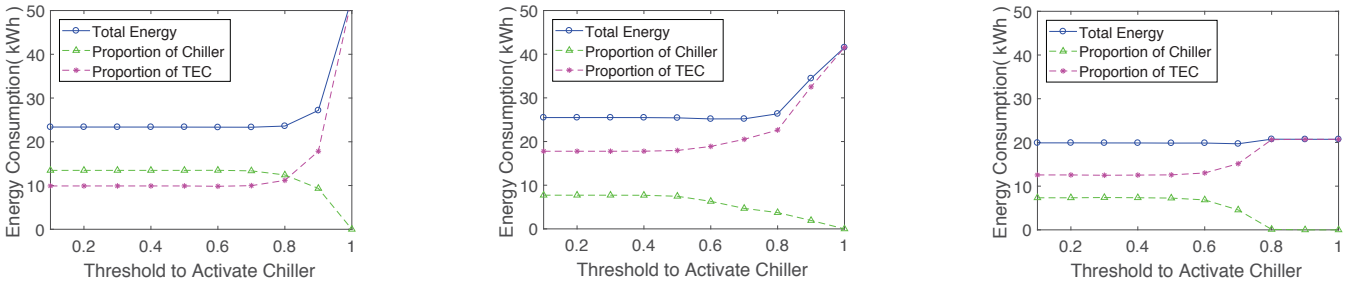
Energy saving compared with different strategies: Fig 14 shows the energy consumption to handle the cooling mismatching by different strategies. As we can see, compared with conventional water cooling ($E_{chiller}$), our hybrid water cooling (E_{hybrid}) can reduce the cooling energy by 58.72%, 74.48%, 78.43% for *drastic*, *irregular* and *common* workloads, respectively.

Usually, the energy consumption of CPU grows linearly with its utilization [20, 52]. The maximum power consumption of Xeon E5-2650 V3 in our evaluation is 105 W [56]. Then we can roughly calculate the energy consumption is 435.43 kWh for *drastic* trace of 1313 machines in 12 hours (average utilization: 26.32%), 494.42 kWh for the *irregular* trace of 1000 machines in 24 hours (average utilization: 19.62%) and 463.68 kWh for the *common* trace of 1000 machines in 24 hours (average utilization: 18.40%), respectively. We



(a) Drastic (b) Irregular (c) Common

Figure 15: Energy consumption to handle cooling mismatching with different warm water temperature settings (Pct = 80%, cooling control interval is 15 minutes).



(a) Drastic (b) Irregular (c) Common

Figure 16: Energy consumption to handle cooling mismatching with different threshold settings (T_{water} = 50°C, cooling control interval is 15 minutes).

define a partial Power Usage Effectiveness (pPUE) [8] as:

$$pPUE = \frac{CPU\ Energy + CPU\ Cooling\ Energy}{CPU\ Energy}. \quad (6)$$

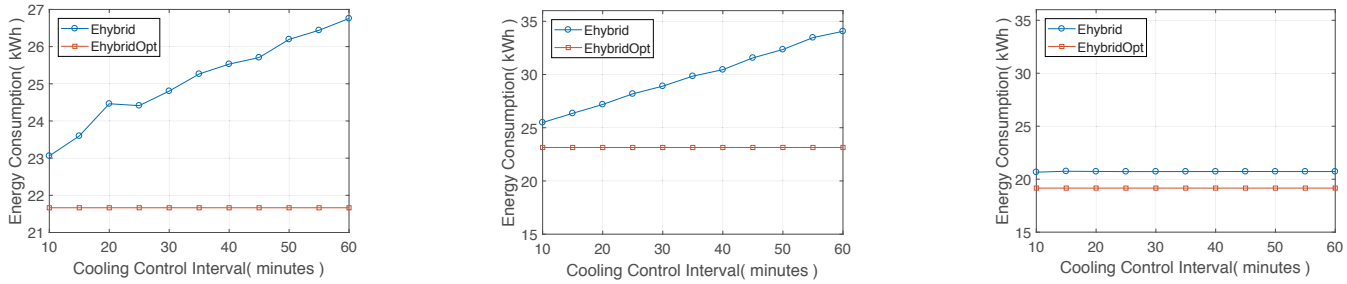
Then according to the cooling energy shown in Fig. 14, we can calculate that pPUE of 50℃ warm water cooling in a conventional water cooling system is 1.13, 1.21, 1.21 for the three traces and our hybrid water cooling can achieve pPUE of 1.05, 1.05 and 1.04.

Next, we discuss the impact of the three settings (including the water temperature, the length of cooling control interval and the threshold to activate chiller) on the energy efficiency of the adaptive cooling control.

Impact of the water temperature: the temperature of the water from cooling tower depends on the outside air temperature. Hot weather means higher water temperature. Fig. 15 shows the cooling energy comparison when the water temperature from cooling tower grows. Due to the temperature growth of supplied water, each CPU's temperature also increases, which leads to more frequent usage of chiller/TEC. In Fig. 15(a), E_{tec} becomes larger than $E_{chiller}$, this is because we adopt the worst cooling case of TEC and its energy efficiency (i.e., temperature reduction/watt) becomes worse when handling more temperatures as shown in Fig 10. In Fig. 15(b), we can see E_{TEC} and E_{hybrid} are the same when water temperature is lower than 48℃. This is because the percentage of hot spots is lower our setting 80%, and chiller is not triggered. When the water temperature is larger than 48℃ , the percentage of hot spots increases and it triggers hybrid cooling, which achieves a better cooling efficiency.

Impact of the threshold to activate chiller: Fig. 16 shows the energy consumption with different threshold settings. Setting a threshold larger than 0.8 would incur more energy consumption to handle the cooling mismatching. This is because chiller can hardly be activated and only TEC is working. As we can see, 0.8 is a suitable upper threshold for different workloads to achieve better energy efficiency. Smaller threshold only cause a little increase in energy consumption. This is because when the percentage of hot spots is low, its average CPU temperature is low. The cooling capacity provided by chiller is based one the minimum average CPU temperature in our adaptive cooling control. Even though chiller is activated, the cooling capacity provided by chiller is small. In this way, we can avoid overcooling some servers and energy waste. So this feature makes our cooling control more robust to the small threshold.

Impact of the length of the cooling control interval: this is another parameter that may affect the cooling energy efficiency. Fig. 17 shows the energy consumption to handle the cooling mismatching with different cooling control interval settings. As we can see, a smaller cooling control interval can achieve better energy efficiency. This is because our cooling control mechanism is based on the cooling demands in the previous interval. Setting a smaller



(a) Drastic (b) Irregular (c) Common

Figure 17: Energy consumption to handle cooling mismatching with different cooling control interval settings (T_{water} = 50°C, Pct = 80%).

interval can update the cooling strategy more often to follow the cooling demand's variations. But if we have pre-knowledge of the cooling demand, then the length of the cooling control interval has no impact on the energy efficiency (see $E_{hybridOpt}$), because we can always figure out the optimal decisions in advance. Especially, we can see the time control interval has more impact on *drastic* and *irregular* workload. This is because these two workloads both have significant utilization variations. From Fig. 13(c), we can see only a small period would trigger hybrid cooling for *common* workload. So cooling control interval has little impact on it. Overall, a smaller cooling control interval can achieve better energy efficiency in our proposed cooling framework.

8 RELATED WORK

Thermal management in micro-architectures: due to the increasing transistor density, the cooling limitation has become a key factor that hinders the performance improvement of chips. [27] proposes a framework to improve the energy efficiency of a chip, while managing the thermal problem at the same time. By reducing power conversion loss throughout execution, ThermoGater [32] mitigates regulator-induced thermal emergencies and achieves a small thermal gradient inside a processor chip. Other works also explore the temperature management in memory, e.g., thermal management of ReRAM [11] and enhancing vertical thermal conduction in processor-memory stacks [1]. Differently, our work explores the cooling management across CPUs at the datacenter level.

Cooling management in datacenters: previous works have proposed cooling energy accounting[29], energy-aware spatial placement of workloads (e.g., [2, 10, 42]), temporal scheduling (e.g., [36] and even geographical load balancing [34] to reduce the cooling energy consumption in datacenters. CoolProvision [40] proposes an under-provision cooling for outside air ("free") cooling by leveraging workload deferral, consolidation and throttling via CPU frequency scaling. [49] proposed thermal time shifting for air cooling in datacenters. It uses phase change materials (PCM) to absorb the heat by melting to shave the peak cooling demands. To make peak shaving more controllable, the authors further propose to control the melting of PCM by workload placement [50]. Different from previous works above, we explore the cooling energy optimization in the emerging water cooling system, and our solution can optimize the water cooling without changing the current workload management strategies. [25, 44] propose a software suite to emulate or predict temperatures for thermal management. Its primary goal is to improve hardware reliability [47] rather than cooling energy optimization. In a computing system, the high temperature or temperature variation has a significant impact on the reliability of hard disk [19, 43, 46]. Especially, hard disks are exposed to high temperature in a "free" cooled datacenter [22]. To overcome the challenge, CoolAir [22] proposes a system for managing temperature variation in "free" cooled datacenters by spatial placement and temporal scheduling of workload. Our work studies the thermal management of CPUs in a datacenter. Different from hard disk, CPU is more resistant to temperature variation [14].

Energy management in datacenters: Many efforts have been paid to provide energy monitoring at different levels (e.g., virtual machine-level[28, 31], server-level[20, 52]) in a datacenter. Based on the results of energy monitoring, power capping strategy is proposed to reduce capital expenses of the power infrastructure in a datacenter by improving the utilization of power infrastructures. SmoothOperator [26] proposes a workload-aware service placement strategy to reshape the power usage of servers and mine the potential power capacity. Other works also investigate increasing the server densities in a datacenter and use distributed batteries and super capacitors at the server level to make up the power mismatching [24, 33, 35]. The hybrid water cooling technique we explore in this paper is inspired by the prior works above.

9 CONCLUSION

In this paper, we propose a hybrid water cooling design, which enables safe warm water cooling to reduce the usage of chillers. To further improve the energy efficiency, we design an adaptive cooling control mechanism, allowing the hybrid water cooling system to efficiently and economically handle the cooling mismatching according to workload variations. Our evaluation based on a hardware prototype and cluster traces from Google and Alibaba demonstrates that our hybrid water cooling system can reduce the energy consumption by 58.72%~78.43% to handle the cooling mismatching, compared to conventional water cooling system.

REFERENCES

[1] Aditya Agrawal, Josep Torrellas, and Sachin Idgunji. 2017. Xylem: Enhancing Vertical Thermal Conduction in 3D Processor-memory Stacks. In *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-50 '17)*. ACM, New York, NY, USA, 546–559. https://doi.org/10.1145/3123939.3124547

[2] Faraz Ahmad and T. N. Vijaykumar. 2010. Joint Optimization of Idle and Cooling Power in Data Centers While Maintaining Response Time. In *Proceedings of the Fifteenth Edition of ASPLOS on Architectural Support for Programming Languages and Operating Systems (ASPLOS XV)*. ACM, New York, NY, USA, 243–256. https://doi.org/10.1145/1736020.1736048

[3] Alibaba. 2018. Alibaba cluster workload traces. https://github.com/alibaba/clusterdata[Online Accessed, 18-Mar-2019].

[4] Alibaba. 2019. Price of thermoelectric cooler. https://www.alibaba.com/product-detail/Thermoelectric-Cooler-12V-61W-40-40_60749472639.html?spm=a2700.7724838.2017115.78.67b32476eA3zoj[Online Accessed, 18-Mar-2019].

[5] Mohammad Alizadeh, Tom Edsall, Sarang Dharmapurikar, Ramanan Vaidyanathan, Kevin Chu, Andy Fingerhut, Vinh The Lam, Francis Matus, Rong Pan, Navindra Yadav, and George Varghese. 2014. CONGA: Distributed Congestion-aware Load Balancing for Datacenters. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14)*. ACM, New York, NY, USA, 503–514. https://doi.org/10.1145/2619239.2626316

[6] ASETEK. 2019. Newsroom. https://www.asetek.com/press-room/news/[Online Accessed, 18-Mar-2019].

[7] ASHRAE. 2015. ASHRAE Thermal Guidelines. https://datacenters.lbl.gov/sites/all/files/ASHRAE%20Thermal%20Guidelines_%20SVLG%202015.pdf[Online Accessed, 18-Mar-2019].

[8] Victor Avelar, Dan Azevedo, Alan French, and Emerson Network Power. 2012. PUE: a comprehensive examination of the metric. *White paper* 49 (2012).

[9] AWS. 2019. Amazon EC2 Instance Types. https://aws.amazon.com/ec2/instance-types/[Online Accessed, 18-Mar-2019].

[10] Cullen Bash and George Forman. 2007. Cool Job Allocation: Measuring the Power Savings of Placing Jobs at Cooling-efficient Locations in the Data Center. In *2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference (ATC'07)*. USENIX Association, Berkeley, CA, USA, Article 29, 6 pages. http://dl.acm.org/citation.cfm?id=1364385.1364414

[11] Majed Valad Beigi and Gokhan Memik. 2018. Thermal-aware Optimizations of reRAM-based Neuromorphic Computing Systems. In *Proceedings of the 55th Annual Design Automation Conference (DAC '18)*. ACM, New York, NY, USA, Article 39, 6 pages. https://doi.org/10.1145/3195970.3196128

[12] Muhammad Tayyab Chaudhry, Teck Chaw Ling, Atif Manzoor, Syed Asad Hussain, and Jongwon Kim. 2015. Thermal-Aware Scheduling in Green Data Centers. *ACM Comput. Surv.* 47, 3, Article 39 (Feb. 2015), 48 pages. https://doi.org/10.1145/2678278

[13] Shutong Chen, Zhi Zhou, Fangming Liu, Zongpeng Li, and Shaolei Ren. 2018. CloudHeat: An Efficient Online Market Mechanism for Datacenter Heat Harvesting. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 3, 3, Article 11 (June 2018), 31 pages. https://doi.org/10.1145/3199675

[14] Henry Coles, Michael Ellsworth, and David J. Martinez. 2011. "Hot" for Warm Water Cooling. In *State of the Practice Reports (SC '11)*. ACM, New York, NY, USA, Article 17, 10 pages. https://doi.org/10.1145/2063348.2063371

[15] CoolIT. 2017. 4 key benefits of using liquid to cool data centers. https://www.coolitsystems.com/data-center-cooling-power-myths-busted-2/ [Online Accessed, 18-Mar-2019].

[16] CoolIT. 2017. Centralized vs. Distributed Pumping for Rack-based Direct Liquid Cooling. https://www.coolitsystems.com/wp-content/uploads/2017/07/coolit_centralized_vs_distributed_pumping_rev08.pdf[Online Accessed, 18-Mar-2019].

[17] CoolIT. 2018. Case study. https://www.coolitsystems.com/pressroom/case-studies/[Online Accessed, 18-Mar-2019].

[18] CoolIT. 2018. Rack DCLC Product Guide. https://www.coolitsystems.com/wp-content/uploads/2018/11/771-00012-rev-a18_new-coolit-rack-dclc-product-guide.pdf[Online Accessed, 18-Mar-2019].

[19] Nosayba El-Sayed, Ioan A. Stefanovici, George Amvrosiadis, Andy A. Hwang, and Bianca Schroeder. 2012. Temperature Management in Data Centers: Why Some (Might) Like It Hot. (2012), 163–174. https://doi.org/10.1145/2254756.2254778

[20] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power Provisioning for a Warehouse-sized Computer. In *Proceedings of the 34th Annual International Symposium on Computer Architecture (ISCA '07)*. ACM, New York, NY, USA, 13–23. https://doi.org/10.1145/1250662.1250665

[21] Maurizio Frizziero. 2016. Rethinking Chilled Water Temperatures Can Bring Big Savings in Data Center Cooling. https://blog.schneider-electric.com/datacenter/2016/08/17/water-temperatures-data-center-cooling/[Online Accessed, 18-Mar-2019].

[22] Íñigo Goiri, Thu D. Nguyen, and Ricardo Bianchini. 2015. CoolAir: Temperature- and Variation-Aware Management for Free-Cooled Datacenters. In *Proceedings of the Twentieth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '15)*. ACM, New York, NY, USA, 253–265. https://doi.org/10.1145/2694344.2694378

[23] Google. 2019. Google cluster workload traces. https://github.com/google/cluster-data[Online Accessed, 18-Mar-2019].

[24] Sriram Govindan, Di Wang, Anand Sivasubramaniam, and Bhuvan Urgaonkar. 2013. Aggressive Datacenter Power Provisioning with Batteries. *ACM Trans. Comput. Syst.* 31, 1, Article 2 (Feb. 2013), 31 pages. https://doi.org/10.1145/2427631.2427633

[25] Taliver Heath, Ana Paula Centeno, Pradeep George, Luiz Ramos, Yogesh Jaluria, and Ricardo Bianchini. 2006. Mercury and Freon: Temperature Emulation and Management for Server Systems. In *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS XII)*. ACM, New York, NY, USA, 106–116. https://doi.org/10.1145/1168857.1168872

[26] Chang-Hong Hsu, Qingyuan Deng, Jason Mars, and Lingjia Tang. 2018. SmoothOperator: Reducing Power Fragmentation and Improving Power Utilization in Large-scale Datacenters. In *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '18)*. ACM, New York, NY, USA, 535–548. https://doi.org/10.1145/3173162.3173190

[27] Michael Huang, Jose Renau, Seung-Moon Yoo, and Josep Torrellas. 2000. A Framework for Dynamic Energy Efficiency and Temperature Management. In *Proceedings of the 33rd Annual ACM/IEEE International Symposium on Microarchitecture (MICRO 33)*. ACM, New York, NY, USA, 202–213. https://doi.org/10.1145/360128.360149

[28] Weixiang Jiang, Fangming Liu, Guoming Tang, Kui Wu, and Hai Jin. 2017. Virtual Machine Power Accounting with Shapley Value. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. Atlanta, GA, USA, 1683–1693. https://doi.org/10.1109/ICDCS.2017.235

[29] Weixiang Jiang, Shaolei Ren, Fangming Liu, and Hai Jin. 2018. Non-IT Energy Accounting in Virtualized Datacenter. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. Vienna, Austria, 300–310. https://doi.org/10.1109/ICDCS.2018.00038

[30] Ki-Dong Kang, Mohammad Alian, Daehoon Kim, Jaehyuk Huh, and Nam Sung Kim. 2018. VIP: Virtual Performance-State for Efficient Power Management of Virtual Machines. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC '18)*. ACM, New York, NY, USA, 237–248. https://doi.org/10.1145/3267809.3267816

[31] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, and Arka A. Bhattacharya. 2010. Virtual Machine Power Metering and Provisioning. In *Proceedings of the 1st ACM Symposium on Cloud Computing (SoCC '10)*. ACM, New York, NY, USA, 39–50. https://doi.org/10.1145/1807128.1807136

[32] S. Karen Khatamifard, Longfei Wang, Weize Yu, Selçuk Köse, and Ulya R. Karpuzcu. 2017. ThermoGater: Thermally-Aware On-Chip Voltage Regulation. In *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA '17)*. ACM, New York, NY, USA, 120–132. https://doi.org/10.1145/3079856.3080250

[33] Vasileios Kontorinis, Liuyi Eric Zhang, Baris Aksanli, Jack Sampson, Houman Homayoun, Eddie Pettis, Dean M. Tullsen, and Tajana Simunic Rosing. 2012. Managing Distributed Ups Energy for Effective Power Capping in Data Centers. In *Proceedings of the 39th Annual International Symposium on Computer Architecture (ISCA '12)*. IEEE Computer Society, Washington, DC, USA, 488–499. http://dl.acm.org/citation.cfm?id=2337159.2337216

[34] Kien Le, Ricardo Bianchini, Jingru Zhang, Yogesh Jaluria, Jiandong Meng, and Thu D. Nguyen. 2011. Reducing Electricity Cost Through Virtual Machine Placement in High Performance Computing Clouds. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC '11)*. ACM, New York, NY, USA, Article 22, 12 pages. https://doi.org/10.1145/2063384.2063413

[35] Longjun Liu, Chao Li, Hongbin Sun, Yang Hu, Juncheng Gu, Tao Li, Jingmin Xin, and Nanning Zheng. 2015. HEB: Deploying and Managing Hybrid Energy Buffers for Improving Datacenter Efficiency and Economy. In *Proceedings of the 42Nd Annual International Symposium on Computer Architecture (ISCA '15)*. ACM, New York, NY, USA, 463–475. https://doi.org/10.1145/2749469.2750384

[36] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. 2012. Renewable and Cooling Aware Workload Management for Sustainable Data Centers. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '12)*. ACM, New York, NY, USA, 175–186. https://doi.org/10.1145/2254756.2254779

[37] Power Knot LLC. 2011. COPs, EERs, and SEERs How Efficient is Your Air Conditioning System? https://www.powerknot.com/wp-content/uploads/sites/6/2011/03/Power_Knot_about_COP_EER_SEER.pdf[Online Accessed, 18-Mar-2019].

[38] David Lo, Liqun Cheng, Rama Govindaraju, Luiz André Barroso, and Christos Kozyrakis. 2014. Towards Energy Proportionality for Large-scale Latency-critical Workloads. In *Proceeding of the 41st Annual International Symposium on Computer Architecuture (ISCA '14)*. IEEE Press, Piscataway, NJ, USA, 301–312. http://dl.acm.org/citation.cfm?id=2665671.2665718

[39] Chengzhi Lu, Kejiang Ye, Guoyao Xu, Cheng-Zhong Xu, and Tongxin Bai. 2017. Imbalance in the cloud: An analysis on Alibaba cluster trace. In *2017 IEEE International Conference on Big Data (Big Data)*. 2884–2892. https://doi.org/10.1109/BigData.2017.8258257

[40] Ioannis Manousakis, Íñigo Goiri, Sriram Sankar, Thu D. Nguyen, and Ricardo Bianchini. 2015. CoolProvision: Underprovisioning Datacenter Cooling. In *Proceedings of the Sixth ACM Symposium on Cloud Computing (SoCC '15)*. ACM, New York, NY, USA, 356–367. https://doi.org/10.1145/2806777.2806938

[41] TEC Microsystems. 2019. TE Cooler Reliability Testing and Reports. https://www.tec-microsystems.com/faq/reliability-testing-and-reports.html[Online Accessed, 18-Mar-2019].

[42] Justin Moore, Jeff Chase, Parthasarathy Ranganathan, and Ratnesh Sharma. 2005. Making Scheduling "Cool": Temperature-aware Workload Placement in Data Centers. In *Proceedings of the Annual Conference on USENIX Annual Technical Conference (ATEC '05)*. USENIX Association, Berkeley, CA, USA, 5–5. http://dl.acm.org/citation.cfm?id=1247360.1247365

[43] Eduardo Pinheiro, Wolf-Dietrich Weber, and Luiz André Barroso. 2007. Failure Trends in a Large Disk Drive Population. In *Proceedings of the 5th USENIX Conference on File and Storage Technologies (FAST '07)*. USENIX Association, Berkeley, CA, USA, 2–2. http://dl.acm.org/citation.cfm?id=1267903.1267905

[44] Luiz Ramos and Ricardo Bianchini. 2008. C-Oracle: Predictive thermal management for data centers. In *2008 IEEE 14th International Symposium on High Performance Computer Architecture*. 111–122. https://doi.org/10.1109/HPCA.2008.4658632

[45] Data Center Research. 2019. Data Center Map. https://www.datacentermap.com/[OnlineAccessed,18-Mar-2019]

[46] Sriram Sankar, Mark Shaw, Kushagra Vaid, and Sudhanva Gurumurthi. 2013. Datacenter Scale Evaluation of the Impact of Temperature on Hard Disk Drive Failures. *Trans. Storage* 9, 2, Article 6 (July 2013), 24 pages. https://doi.org/10.1145/2491472.2491475

[47] Osman Sarood, Esteban Meneses, and Laxmikant V. Kale. 2013. A 'Cool' Way of Improving the Reliability of HPC Machines. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC '13)*. ACM, New York, NY, USA, Article 58, 12 pages. https://doi.org/10.1145/2503210.2503228

[48] Yizhou Shan, Yutong Huang, Yilun Chen, and Yiying Zhang. 2018. LegoOS: A Disseminated, Distributed OS for Hardware Resource Disaggregation. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. USENIX Association, Carlsbad, CA, 69–87. https://www.usenix.org/conference/osdi18/presentation/shan

[49] Matt Skach, Manish Arora, Chang-Hong Hsu, Qi Li, Dean Tullsen, Lingjia Tang, and Jason Mars. 2015. Thermal Time Shifting: Leveraging Phase Change Materials to Reduce Cooling Costs in Warehouse-scale Computers. In *Proceedings of the 42Nd Annual International Symposium on Computer Architecture (ISCA '15)*. ACM, New York, NY, USA, 439–449. https://doi.org/10.1145/2749469.2749474

[50] Matt Skach, Manish Arora, Dean Tullsen, Lingjia Tang, and Jason Mars. 2018. Virtual Melting Temperature: Managing Server Load to Minimize Cooling Overhead with Phase Change Materials. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*. 15–28. https://doi.org/10.1109/ISCA.2018.00013

[51] Moritz Steiner, Bob Gaglianello Gaglianello, Vijay Gurbani, Volker Hilt, W.D. Roome, Michael Scharf, and Thomas Voith. 2012. Network-aware Service Placement in a Distributed Cloud Environment. *SIGCOMM Comput. Commun. Rev.* 42, 4 (Aug. 2012), 73–74. https://doi.org/10.1145/2377677.2377687

[52] Guoming Tang, Weixiang Jiang, Zhifeng Xu, Fangming Liu, and Kui Wu. 2015. Zero-Cost, Fine-Grained Power Monitoring of Datacenters Using Non-Intrusive Power Disaggregation. In *Proceedings of the 16th Annual Middleware Conference (Middleware '15)*. ACM, New York, NY, USA, 271–282. https://doi.org/10.1145/2814576.2814728

[53] The Engineering ToolBox. 2019. Thermal Conductivity of common Materials and Gases. https://www.engineeringtoolbox.com/thermal-conductivity-d_429.html[Online Accessed, 18-Mar-2019].

[54] Neven Ukrainczyk, Stanislav Kurajica, and Juraj Šipušić. 2010. Thermophysical comparison of five commercial paraffin waxes as latent heat storage materials. *Chemical and biochemical engineering quarterly* 24, 2 (2010), 129–137.

[55] CPU World. 2018. Minimum Maximum operating temperatures. http://www.cpu-world.com/Glossary/M/Minimum_Maximum_operating_temperatures.html[Online Accessed, 18-Mar-2019].

[56] CPU World. 2019. Intel Xeon E5-2650 v3 specifications. http://www.cpu-world.com/CPUs/Xeon/Intel-Xeon%20E5-2650%20v3.html[Online Accessed, 18-Mar-2019].

[57] Kaicheng Zhang, Akhil Guliani, Seda Ogrenci-Memik, Gokhan Memik, Kazutomo Yoshii, Rajesh Sankaran, and Pete Beckman. 2018. Machine Learning-Based Temperature Prediction for Runtime Thermal Management Across System Components. *IEEE Transactions on Parallel and Distributed Systems* 29, 2 (Feb 2018), 405–419. https://doi.org/10.1109/TPDS.2017.2732951