An Adaptive Approach for Dealing with Flow Disruption in Virtualized Water-Cooled Data Centers

Udaya Puvvadi, Anuroop Desu, Tyler Stachecki, Kanad Ghose, Bahgat Sammakia
{upuvvad1, adesu1, tstache1, ghose, bahgat}@binghamton.edu
State University of New York, Binghamton NY 13902

***Abstract*—The recent availability of water cooling systems that can be easily retrofitted to stock servers by replacing the heatsinks with coldplates has made it possible to use such systems for non-HPC cloud/data center servers. These cooling systems use pumps to circulate water and the pumps are likely to fail in the long run. We present a technique to handle flow disruptions caused by the pump failures in a virtualized environment. The solution uses an estimation of the residual cooling capacity left in the failed cooling system to adaptively adjust the CPU clock frequency as virtual machines are migrated off the racks affected by the failure. This minimizes the degradation of the tail latencies of the served requests during the migration interval for all servers affected by the failure, as seen in the experimental results.**

Keywords: water-cooled servers; virtual machines; dependable systems; virtual machine migration; gang migration

I. Introduction

Direct liquid cooling systems (DLCS) have appeared recently in the market at competitive pricing levels. DLCS brings liquid directly to the hottest components inside a server such as the CPU chips and DRAM DIMMs. These DLCS, including solutions from Asetek [1] and CoolIT [2], use coldplates to replace CPU and DRAM heatsinks and circulate water at normal environmental temperatures ("warm water") or chilled water through these cold plates to take out the dissipated heat. Water with its high thermal conductivity can be more effective at removing heat compared to air-cooling solutions. Other liquid cooling solutions include racks with rear door heat exchangers that circulate chilled water, in-line coolers and immersion cooling systems that immerse IT equipment in electrically inert fluids [5].

II. Backgroud

In a typical DLCS, each server has two connections for circulating water: one to bring in the warm water from a rack-level heat exchanger (RHX) via a supply manifold and another to take out the heated water from the coldplates back to the RHX via a return manifold, as shown in Figures 1 and 2. The supply manifold, the coldplates and the connections within the RHX form a loop (called the **coldplate loop, C-loop**). Another loop (called the **facility-side loop, F-loop**) runs the chilled or "warm" water from the facility side (blue line) and uses the RHX to transfer the heat from the C-loop to an evaporative cooling tower (via the red-colored line) and thus to the environment. Both of these loops contain pumps, that control the flow rate of the water being supplied to maintain an optimum heat exchange. Fig. 2, shows a configuration of an RHX, the CoolIT DCLC CHx 40 unit used in our studies, capable of handling three adjacent server racks using three sets of manifolds in parallel. Fig. 2 depicts the flow paths within server coldplates and within the RHX in a rack. In the DLCS system used for this study, the chilled water supply manifold includes a single pump ("P" in Fig. 2) to circulate water in the F-loop to all heat exchangers in a row/aisle. Inside each rack-level heat exchanger, two pumps ("S1" and "S2") are used in series for fault-tolerance within the C-loop.

III. Failure Modes And Dealing with Failures

The mechanical pumps that are installed in both the F-loop and in the C-loop are likely to fail in long run and can disrupt the cooling for a server.

Complete Flow Disruption in the F-Loop: We first consider a scenario where the pump in the F-loop ("P" in Fig. 2) fails, resulting in shutting down the circulation in the F-loop. However, the C-loop continues to run, and circulate the water through the coldplates and the RHX. As a consequence of the F-loop failure, progressively lower amounts of heat will be taken off the C-loop by the rack-mounted heat exchanger, increasing the water temperature within in the C-loop. If the server activities continue,

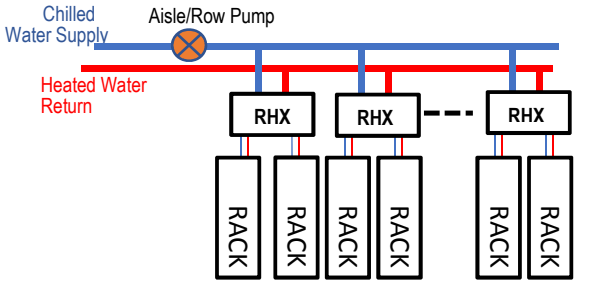

Figure 1. Aisle/Row pump and rack-scale shared heat exchangers (**HE**s) – two racks share a HE in this example

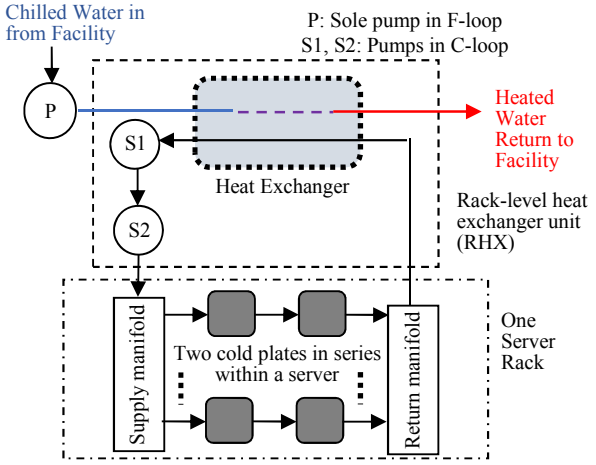

Figure 2. Water flow paths and pump locations

2159-6190/19/$31.00 ©2019 IEEE
DOI 10.1109/CLOUD.2019.00057

increasing CPU core temperatures will first slow down the CPU clock (using dynamic voltage and frequency scaling, DVFS), to reduce power dissipation, and ultimately shuts down the servers when core temperatures exceed safe limits.

Fig. 3 (a) depicts what happens when the F-loop pump fails, and a rack of 16 servers connected to the system are running synthetic applications that exercise the cores at 100% utilization level at 100% clock rate. When the F-loop circulation failure is simulated at time 1000 Secs, by closing off the F-loop's inlet valve to the RHX, the CPU core temperatures go up steadily, but slowly, as shown. However, the core temperature increase is not enough to hit the CPU threshold temperature (around 90 degrees Celsius) at which the CPU clock throttling commences. Instead, the core temperature goes up following the F-loop pump failure to a level where the server fans run at full speed (about 15,000 rpm) and stays there (Fig. 3 (b)). As the coldplates are encased in plastic jackets, the server-internal fans only provide indirect and marginal cooling to the CPU by taking away the heat that is conducted to the motherboard via the CPU pins. In spite of this, as seen in Fig. 3 (b), server fans ramp up to full speed to cool the CPU on sensing an increased core temperature, failing to counter the steady but slow increase in the CPU core temperature induced by the F-loop flow loss.

Starting from the time of induced failure (at 1000 Secs.) to the time till the fans run at full speed (at 2189 Secs.), that is, for an interval of 1189 Secs. (roughly 19.8 minutes), residual cooling in the DCLS provides cooling to the CPU before the server fans ramp up to full speed. This duration is referred to as the **actionable time**. Residual cooling primarily comes from the high heat capacity of the cold water left in the F-loop of the heat exchanger, specifically within its internal tank, the two manifolds and all plumbing lines on both loops.

Running fans in the servers at their maximum speed has various downsides, some of them are: (a) fans running at full speed introduce severe wear on the fan bearings, reducing their lifetime dramatically [10]; (b) power is wasted (typically, 6 Watts at minimum speed vs. 14 Watts at full speed per server fan); (c) fans running at full speed can cause back pressure inside the server cabinet and recirculate hot air within the server, as demonstrated experimentally in [6].

From Fig. 3 (a), the residual cooling capacity (**RCC**) available in the system for running and migrating VMs before fans ramp up to their full speed is ~2746 KJ. This is calculated based on the power drawn (roughly) by the 16 servers in the rack during the period 1000 to 2189 Seconds, dissipating 123 to 131 Watts for both sockets during this period. This RCC value is used in deciding when DVFS has to be used to throttle down the CPU clock, to reduce the demand for residual cooling while limiting the impact on tail latency of the requests being served. When the system is scaled up by connecting 2 adjacent racks to the RHX, with 32 servers per rack, the resulting actionable time is a quarter of the original actionable time (= 19.8/4, i.e., 4.95 minutes) which is observed on a single rack, assuming that the CPUs are

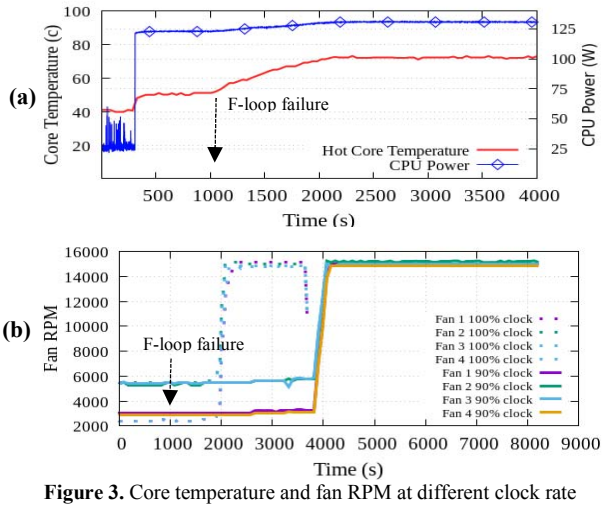

Figure 3. Core temperature and fan RPM at different clock rate

exclusively using the RCC. Similarly, when system is scaled up to cool 3 racks, the actionable time is reduced to 3.3 minutes (19.8/6).

Failures in the C-Loop: The likelihood of failure of two serially connected pumps at the same time in C-loop is highly unlikely. When one of these two pumps fail, the operating pump still maintains a significant flow and thereby it's impact on server activities is minimal. Due to these reasons, C-loop failures are not studied further.

Dealing with F-Loop Pump Failures: Redundant pumps in F-loop can address the failure, however having redundant pumps is expensive because of the costs of plumbing, the inclusion of remotely-controlled flow-diverting valves and pump control system. *We propose a software solution that avoids these additional expenses.*

When the F-loop pump failure occurs, all of the racks connected to the DLCS system are impacted. These racks, the servers within these racks, and virtual machines (**VMs**) hosted on these servers are called **affected racks**, **affected servers** and **affected VMs** respectively. The proposed solution deals with F-loop pump failure in a virtualized environment by migrating the VMs on the affected racks to non-affected racks before the fans in the affected servers ramp up to full speed. VM migration is done in batches, as in [4], to reduce network and NFS server contention. Further, CPU clock slowdown is delayed as much as possible to reduce any impact on the tail latencies of serviced requests. After migrating a batch of VMs, if the migration of the remaining VMs cannot be completed without exceeding the available (instantaneous) RCC, CPU clock is reduced adaptively using the DVFS.

IV. Cooling-Constrained Adaptive VM Migration

This section describes the algorithm used to migrate VMs on flow disruptions in the F-loop cooling failure.

A. Goals and Overview

The goals of the adaptive algorithm are:

1. The cooling capacity left for the affected hosts is limited and must be used to migrate as many VMs off the affected hosts as possible.
2. Services running on affected VMs should experience as little delay as possible: tail latency increases due to VM migration/throttling of the CPU clock in the upper 95th percentile must be limited for as many services as possible that are running on the affected VMs.

These goals have conflicting requirements. Running the CPUs on the affected servers at maximum frequency will potentially limit the performance impact but may quickly exhaust the RCC, precluding the migration of all affected VMs. The adaptive migration algorithm presented here attempts to migrate as many VMs as possible at the highest CPU clock frequency and reduces the CPU clock rate during the migration of the remaining VMs to stay within the cooling budget left. The algorithm does this by pipelining the migration of VMs concurrently in small batches and estimating the instantaneous RCC left after a VM batch is migrated to decide if clock throttling is needed to migrate the remaining affected VMs. When all VMs have been migrated off an affected server, the server is powered off.

Our technique requires an estimation of the residual cooling left after a batch of VM is migrated off the affected servers. This estimation requires the CPU power readings to be obtained from the server's power management unit (PMU) to determine the power (and residual cooling) consumed by running VMs and VMs migrated in a batch.

B. Systems Architecture and Implementation

All VM management functions are implemented in a Front-End Scheduler (FES) which has three concurrently executing threads that work independently:

- The *Energy Estimation Thread (EET)* gathers the power consumption data from all the servers at intervals of 0.5 seconds and performs estimation of energy used and future energy needs by integrating CPU power over the time interval.
- The *Commands Thread (CT)* is responsible for sending commands to the servers to adjust the DVFS setting, to trigger VM migration and to power off servers.
- The *Scheduler Thread (ST)*, analyzes the data from servers and flow sensors in the RHX to detect F-loop failure, detect high core temperatures and implement a selection strategy (to be described in Sec. IV. C) to migrate the VMs from affected racks to non-affected racks.

C. VM Selection Strategies

The order of selection of the VMs for migration in batches can affect the migration time. We evaluated three different VM selection strategies for migration of the affected VMs from affected servers to non-affected servers, to understand the impact on the migration times. The first two strategies begin by selecting the VMs from the servers with the highest power dissipation. Intuitively, delaying the migration of VMs from such servers will consume higher amount of residual cooling as they continue to run on the affected server. VMs within these servers are chosen as follows:

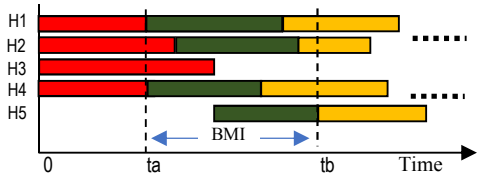

Figure 4: Concurrent VM migration in batches of fours off five hosts with pipelining. VMs belonging to a batch share a common color

HPS-HU: Highest CPU utilization VM from the selected server: The VM selected for migration is the one with the highest CPU usage from the server consuming most power. This VM is responsible for the high CPU power dissipation.

HPS-HM: Highest memory footprint VM first from the selected server: A VM with heavy memory utilization (that is, highest memory footprint) is selected for migration from the server consuming the highest power. This VM is likely to have more dirty pages, requiring a higher migration time.

The other VM selection strategy is agnostic of server's power dissipation and is as follows:

HU: VM with highest CPU utilization first: A VM with highest CPU utilization among all the remaining VMs is selected for migration. The rationale is that such VMs are going to dissipate more energy than their peers.

D. Energy Consumption During the Migration of a Batch

Ideally, VMs are migrated in batches off hosts (servers), and all VM migrations within a batch are expected to complete simultaneously, before the next batch of VMs are migrated. This is not the case in reality, as VMs will be running different workloads and their migration times will differ from other VMs in the same batch. Waiting for the all VMs from a single batch to migrate will result in idling and will prolong the overall migration time for the affected VMs, so migration is pipelined as shown in Fig. 4.

VMs are migrated in batches of B virtual machines to avoid resource contention. For each batch of VMs migrated, we define a Batch Migration Interval (**BMI**) that begins with the initiation of the first VM migration within the batch and ends when the last VM in the batch has completed migration. As soon as a VM migration is completed, the migration of a VM from the next batch is started. For a specific BMI, say the k-th interval, BMI (k), we define the following:

(a) E_r: energy spent by running VMs that are not migrated;
(b) E_m: energy spent in migrating the B VMs in the batch;
(c) E_n: energy spent in running VMs from the next batch.

The integration of the server power obtained from the PMU over the BMI, that is the energy spent within the BMI, say, E (k) = $E_r + E_m + E_n$.

Assuming that P_r and P_m are the power dissipated by a running VM and by a migrating VM, respectively, on the average, and T(k) as the duration of BMI (k), we get:

$$E(k) = R(k) * P_r * T(k) + B * P_m * T(k) + E_n$$

where R(k) is the number of running VMs in BMI (k). As an approximation, if E_n is assumed to be the same for all BMIs, we can measure E(k) for three consecutive intervals, E(k), E(k-1) and E(k-2) and solve for the three unknowns (P_r, P_m and E_n). In our technique, the values for P_r, P_m and E_n are continuously re-estimated to reduce approximation errors.

Once P_r, P_m and E_n are evaluated, the residual cooling energy needed to migrate the remaining VMs can be estimated as:

$$CN(k+1) = \Sigma ((R(k) - B) * P_r * T + B * P_m * T + E_n)$$

where the summation is carried out over the remaining batch migration intervals needed to complete all VM migrations and where T is the average BMI duration estimated thus far. The number of remaining BMIs equals $\lceil (R(k) - B)/B \rceil$, since there are B fewer VMs left to migrate at the end of an interval.

The cooling capacity spent just before the commencement of BMI (k+1), say CU(k+1), is clearly the sum of E(k)s from all past intervals, so that the cooling capacity left before BMI(k+1) begins is C0 - CU(k+1), where C0 is the residual cooling capacity (RCC) left immediately after the pump failure in the F-loop (Sec. II). If CN (k+1) > (C0 - CU(k+1)), then sufficient cooling is not available to run the CPUs at the full clock rate, so the CPU clocks are throttled; otherwise, migration continues as before at full clock speed.

Note that our algorithm for adaptive migration is somewhat conservative. When all VMs are migrated off an affected server, it is shut down, saving idling energy. This is not accounted for in the way CN is estimated and compensates for approximations that have been made.

E. Evaluation and Variants

We evaluate two variants of the algorithm, based on when CPU throttling mechanism is triggered: (a) The *Eager Adaptive Migrator* (**EAM**), where the decision to throttle the CPU is based on whether the RCC left is enough to migrate the next batch of VMs alone; (b) The **Baseline**, where the decision to throttle the CPU clock is based on whether RCC left is enough to migrate all the remaining VMs. Thus, compared to Baseline, EAM is likely to migrate more VMs at the highest clock frequency setting.

V. Experimental Assessment

We evaluate our prototype implementation on a set of 2 racks, consisting of 16 servers, with 13 Dell R520 servers, 2 Dell R730 v3 servers and one Dell R730 v4 server, and 64 GB of memory on each server. Of the 2 set of racks, one rack is equipped with warm water cooling system, and other rack uses traditional air cooling system. 10 Gbps network links are used for network connectivity across the servers. CPU power is read off using the RAPL interface. The RCC with this heterogeneous set of 16 servers is estimated by using the lowest possible actionable time of all servers.

A medium sized VM configuration was chosen with 4 VCPUs, and 4GB of memory, and running base Ubuntu 16.04 LTS. All of the VM's disk images were hosted on Network File System (**NFS**) and a total of 94 VMs are hosted. These 94 VMs are divided into 5 virtual groups at load-balancer to run different workloads. Each virtual group contains minimum of one VM from each host and processes different workloads. These are distributed across servers based on CPU and memory capacity with 5 VMs on each Dell R520, 9 VMs on each Dell R730 v3 and 11 VMs on Dell R730 v4. A default qemu migration policy is used on hosts to migrate VMs across servers. A F5 Networks BIG-IP 4000s LTM load-balancer is used to distribute the requests across VMs using least number of connections as balancing policy.

We evaluate two different configurations of cooling systems by scaling the RCC determined from a single rack of 16 servers. The first one uses a scaled up system of **2-Racks,** and the second uses scaled up system of **3-Racks**., as explained in Sec. III. These configurations are studied since the RHX unit used can accommodate the load of up to three scaled-up racks (32 servers/rack) in real scenarios.

Workload: We used five benchmarks, namely, Compress, Crypto.Rsa, Scimark.Monte_Carlo, Serial and Xml.Validation from the SPECjvm2008 suite [9] as workload for our experiments. The requests were served over http, using a workload generator to realize a heterogeneous mix of workload to exercise the servers.

Results: We present 8 different variations of the algorithm, Baseline with (2-Rack), (3-Rack), and EAM with (2-Rack), (3-Rack) configuration, with a batch size (B) of 4, and 8 respectively. The results are shown in Table 1. As expected, choosing the VMs from the higher-powered servers (HPS-HU) permits a higher percentage of VMs to be migrated at higher CPU clock rate, since migrating the VMs of the high-power server's leaves a larger amount of RCC to permit more of the remaining VMs to be migrated at a higher clock rate. Note that after the VMs have been migrated off the three high powered servers (Dell R730s), the difference between the HPS-HU and HU tend to blur as the remaining servers are not distinguishable from each other as they have identical peak power dissipation. Consequently, from the very beginning, the selection of VM for migration is dominated by VMs from servers that have identical peak power dissipation. As a result, there is no consistent winner among the 3 selection strategies. From Table 1, it is seen that when the heat exchanger is used to cool 3 server racks instead of 2 server racks, throttling always happens earlier and a

VARIANT	HPS-HU		HPS-HM		HU	
	% FC	% TC	% FC	% TC	% FC	% TC
BASELINE (2-Rack) 4 VMs per batch	4.3	95.7	4.3	95.7	4.3	95.7
EAM (2-Rack) 4 VMs per batch	64.4	35.6	46.8	53.2	55.3	44.7
BASELINE (3-Rack) 4 VMs per batch	4.3	95.7	4.3	95.7	4.3	95.7
EAM (3-Rack) 4 VMs per batch	25.5	74.5	8.5	91.5	21.2	78.8
BASELINE (2-Rack) 8 VMs per batch	8.5	91.5	8.5	91.5	8.5	91.5
EAM (2-Rack) 8 VMs per batch	48.9	51.1	34	66	60	40
BASELINE (3-Rack) 8 VMs per batch	8.5	91.5	8.5	91.5	8.5	91.5
EAM (3-Rack) 8 VMs per batch	8.5	91.5	17	83	25.5	74.5

Table 1. Percentage of VMs migrated before (%FC – full clock) and after (%TC - throttled) clock throttling for different configurations.

lower percentage of VMs can be moved at the higher clock rate. This is simply because the RCC is shared by more servers. Any increase in the service latencies, particularly any sharp increase in the tail latencies at the 95-th percentile, should be avoided in general to comply with service level agreements. Fig. 5 shows the average 95th percentile tail latencies for the EAM (3-Rack), with 8 VMs per batch. The results are representative of other variants. As seen from Fig. 5, even though throttling occurs for the variants during migration, there is little to no impact on the tail latencies, as migration is IO bound. Thus, one of the original design goals is satisfied. The variations in tail latencies and apparent improvements are all within the variability seen from one run to another. Other VM configurations with different VCPUs and memory show similar results.

VI. Related Work

In [7], the thermal implications of failures in a warm water cooling system on the CPU temperature and air flow were explored. Unfortunately, this study was limited to an affected server load that was grossly below the residual cooling capabilities of the heat exchanger and thus servers were not observed to be throttled, also did not look at service recovery, nor propose any algorithms for migrating the workload. The work presented in this paper addresses these two limitations. In [3], inefficiencies in a warm-water DLCS have been studied but haven't considered any failures. The potential of direct liquid cooling solutions is briefly addressed in [8].

VII. Conclusions

Coldplate-based water cooling systems are emerging for use in stock servers in the non-HPC (High Performance Computing) realm. We examined the implications of failure in such a cooling system and presented a technique for migrating the VMs off affected servers, taking advantage of the residual cooling available. When necessary, the technique throttles the CPU clock to stay within the residual

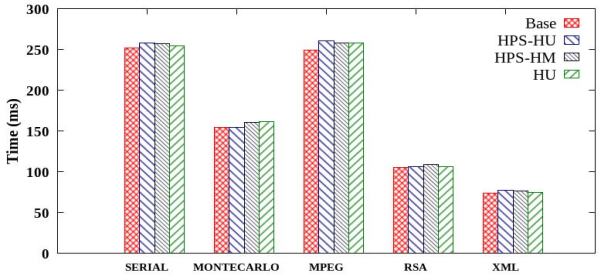

Figure 5: 95th Percentile tail latencies of workloads served in EAM (3-Rack) 8 VMs migration per batch

cooling limits to enable as many VMs as possible to be migrated to other servers. The technique seems to be effective in meeting its goals, as seen in the assessments. When the number of VMs in a batch exceed the number of high-powered servers in the pool of affected servers by a wide margin, the priorities for selecting VMs for migration make little difference. However, when this is not the case, selecting VMs from the servers consuming the highest power for early migration increases the number of VMs migrated without CPU clock throttling, thus minimizing the impact on migration time and its effect on request latency. Finally, reducing the clock frequency by a small amount during migration has little impact on the service latencies as the migrations are fundamentally I/O bounds.

Acknowledgement

This work is supported in part through NSF awards 1040666, 1134867, 1738793 and a NYSERDA award under the PON 2846 competition.

References

[1] Asetek Corpn., Data Center Cooling Solutions Product Pages for Rack DCU and Internal Loop available at http://asetek.com/data-center/data-center-coolers.aspx

[2] CoolIT Systems, Rack DCLC Product Guide, Product Document No. 761-00012, Rev. 04, December 2013.

[3] Coles, Henry *et al*., "Demonstration of Direct Liquid Cooling for Electronic Equipment. California Energy Commission", 2004. available at: http://datacenters.lbl.gov/resources.

[4] Deshpande. U. et al., "Live Gang Migration of Virtual Machines," High-Performance Parallel and Distributed Computing, June 2011.

[5] Direct Liquid cooling Technologies, https://datacenters.lbl.gov/technologies/liquid-cooling

[6] Khalili S, Alissa H, Nemati K, et al. Impact of Internal Design on the Efficiency of IT Equipment in a Hot Aisle Containment System: An Experimental Study. In ASME 2018 InterPACK Conf., August 2018.

[7] S. Alkharabsheh *et al*., "Failure Analysis of Direct Liquid Cooling System in Data Centers" in ACME Journal 2018.

[8] S. Alkharabsheh *et al*., "A Brief Overview of Recent Developments in Thermal Management in Data Centers," Journal of Electronic Packaging, vol. 137, no. 4, p. 040801 (19 pages), 2015.

[9] SPECjvm2008, http://www.spec.org/jvm2008/.

[10] Tian, X., "Cooling fan reliability: failure criteria, accelerated life testing, modeling and qualification", In Proc.. Annual Reliability and Maintainability Symp. 2006, pp. 380-384.