Thermal Behaviors in Liquid Immersion Cooling under Various Workloads: a Case Study

Thomas Randall
Clemson University
Clemson, USA
tlranda@clemson.edu

Bennett Cooper
Clemson University
Clemson, USA
bwc@clemson.edu

Naman Kulshreshtha
Clemson University
Clemson, USA
nkulshr@clemson.edu

Rong Ge
Clemson University
Clemson, USA
rge@clemson.edu

***Abstract*—The growing need for energy-efficient computing has led to many novel system innovations, including liquid immersion cooling. While many myths about the technology have been dispelled, the actual impact of this cooling solution on thermal conditions in real computing scenarios remains under-reported and under-studied. In this work, we collate data from multiple system monitoring tools to perform case-study analyses of the thermal behaviors of immersed hardware, aiming to evaluate the effectiveness of liquid immersion cooling for high-performance and datacenter applications.**

***Index Terms*—Immersion Cooling, Datacenter Computing, Application Study**

I. Introduction

In efforts to improve cooling efficiency, reduce costs, enhance Power Usage Effectiveness (PUE), and ensure hardware reliability, many large-scale computer systems are transitioning from air-cooling solutions to liquid-cooling solutions. Single Phase Liquid Immersion Cooling (SPLIC) is such a technology in which computing hardware is completely submerged in a container filled with a dielectric fluid (coolant). Compared to air, the coolant medium is denser, has higher thermal conductivity and higher heat capacity. These properties relatively improve heat dissipation and permit more efficient heat recycling. Many benefits of SPLIC are well-understood in theory, but the practical effects of the technology on real software workloads have not been well studied academically.

Software has various compute and memory access intensities, stressing processing units and memory systems differently. These hardware components have different power properties and ranges. For example, processing units consume more power and have a larger power range compared to memory devices and disk drives. Component power is also determined by the underlying hardware technologies; for instance, CPUs are generally less power efficient than GPUs, and DDR memory devices are less power efficient than high-bandwidth memory (HBM) devices for the same capacity. Consequently, software induces varying levels of heat dissipation, resulting in different thermal effects within the system. This variation is evident in the spatial temperature map of the system, highlighting areas of high-temperature concentration known as hotspots.

This research was partially supported by the U.S. National Science Foundation under Grant CNS-CCF-1942182.

SPLIC has a different cooling mechanism, control, and operation compared to air cooling. SPLIC dissipates heat in liquid coolant by connecting chilled water tubes passing through the coolant tank. To actively exchange heat, the liquid coolant circulates periodically using pumps or when the coolant temperature exceeds a certain threshold. Unlike air cooling, which directly monitors computer hardware components and responds to their temperatures, SPLIC monitors the temperatures of the coolant entering the pump and the heated chilled water exiting the tank. This separate approach to thermal monitoring inherently delays the system's response to heat accumulation within the hardware. Therefore, understanding the actual impacts of this cooling technology on different hardware and software workloads is important for evaluating its utility and design implications beyond mere hardware configuration.

In this work, we conduct a case study on thermal behaviors in a SPLIC tank with a immersed compute cluster. We utilize several CPU-based and GPU-based High Performance Computing (HPC) and datacenter applications with different compute and memory intensities and analyze hardware thermal behavior and their variations with the SPLIC system configurations and operations. Unlike prior works that rely on simulations, our case study provides empirical insights on thermal demands induced by applications and their effects within the immersed environment.

II. Background & Related Work

One of the major ways to iteratively improve performance through hardware is to increase the device density. However, denser circuitry increases the power density of the components and reduces cooling effectiveness for components such as those deeper within 3D dies, resulting in greater thermal accumulation within the hardware [6]. At the extreme scale, high heat can damage computing hardware and render it inoperable [3]. To prevent such damage, thermal sensors in critical hotspots automatically trigger a decrease in the clock rate of devices to limit additional energy before the device becomes temporarily or permanently inoperable. To ensure performance and throughput, HPC and datacenter systems that push thermal limits require sophisticated cooling solutions. These solutions permit maximal usage for the longest possible period without detrimentally lowering clock rates or causing device failures.

Despite continued innovation in air-cooling technology, air is simply not dense enough to maintain pace dissipating heat from modern HPC and datacenter systems, let alone future designs that are predicted to only increase in Thermal Design Power (TDP). With nearly half of all datacenter energy devoted to cooling hardware and datacenters accounting for 2% of U.S. energy consumption [2], liquid-based cooling technologies are promising and necessary for continued growth and innovation in computing. There are several different approaches to liquid-based cooling, including Direct Liquid Cooling (DLC) and Single- and Dual- Phase Liquid Immersion Cooling (SPLIC and DPLIC, respectively). We limit the focus of our discussion to SPLIC in this work.

A. SPLIC Technology

SPLIC systems include a container with pumps for forced coolant circulation and tubes that can be connected to chilled water sources for heat exchange. Circulation is induced to reach temperature equilibrium of the entire fluid volume more quickly and to facilitate more effective heat exchange. In comparison to Direct Liquid Cooling (DLC), immersion cooling affects all computing components rather than just the areas covered by cold plates where liquid is always flowing. SPLIC systems are more energy-efficient because the pumps only activate periodically to circulate fluid if the coolant temperature exceeds a set point.

B. Prior Evaluations of SPLIC

Many design decisions impact the performance of SPLIC, including the number of pumps, the pump arrangements and settings within the tank, and the choice of dielectric fluid, which have all been studied by many works [3], [4], [9], primarily via simulation or isolated experiments on individual components. Existing work has addressed several initial concerns about the technology, including the extent of dangers to hardware components from absorbing dielectric fluids [8] and the capital burden of procuring, installing and maintaining SPLIC rather than other more traditional cooling systems [7]. However, the existing literature largely lacks evaluations of SPLIC hardware in meeting actual application cooling demands, which we aim to address in this work.

III. Research Questions

We investigate specific questions with real application performance in an SPLIC environment to provide a more comprehensive perspective on the larger subject.

A. RQ1: What are the thermal behaviors of individual hardware components in SPLIC?

Heat accumulates in the pod as power is drawn and used by individual computing components, and generally increases with the hardware's utilization. Most commodity hardware is technically compatible with SPLIC, but have thermal designs that do not explicitly consider properties of ambient liquid coolant. For instance, processors and accelerators typically comprises physical parts such as memory caches and processing cores accommodating both memory access and compute. Such integration improves performance but has a high thermal conductivity and heat exchange within components. SPLIC coolant, if not actively circulated all the time, is unable to reach thermal equilibrium or transfer heat as fast as fan cooling between components in close physical proximity.

It is important to understand if high core temperatures dissipate more heat to proximal components such as memory, particularly for GPUs. Furthermore, the environment's tendency for equilibrium may result in certain hardware components operating at higher temperatures than would be typical given their power design, necessitating cooling interventions for components that may not have DVFS or thermal throttling capabilities.

We observe the actual thermal operating range and rate of temperature change for key hardware components before, during, and after prolonged execution of various applications. These characteristics can be compared relative to the range of temperatures and rate of temperature change in the pod coolant to determine an appropriate response and tolerance within the environment. We also observe the rate of change in hardware hotspot temperatures and coolant temperature to determine if the indirect thermal measurement performed by the cooling system can appropriately respond to high demand from the computing components. We present per-component analyses and coolant-relative per-component analyses in Section V-A.

B. RQ2: How do initial conditions affect SPLIC performance and efficiency?

Some cooling techniques such as DLC yield higher heat dissipation rate as the coolant temperature rises, which is a very convenient upside. We aim to determine whether similar benefits can be observed in SPLIC by correlating heat dissipation with the initial temperature during cooling periods, ensuring that workloads do not interfere with the measurements. We also correlate various component activity measures with temperature dissipation during cooling cycles when applications are running, to determine if different workloads exhibit distinct thermal patterns beyond the rate of heat introduction into the environment. We analyze these trends in Section V-B.

C. RQ3: How do different workloads affect heat accumulation?

Exhaustive studies of application behavior are impractical. In this work, we use a selection of workloads to represent HPC and datacenter applications with similar primary resource demands and study the resulting thermal challenges for the cooling system.

In particular, we consider applications where the primary device used for computation is CPU or GPU and if the computation is subject to a bottleneck in compute or memory throughput. Due to the small number of servers, we exclude network throughput as a bottleneck in our study, though we recognize its importance for large-scale distributed system and workloads. We present a correlative analysis between

workload classification and induced cooling demand in Section V-C.

IV. Experiment Design

We answer the research questions by conducting a set of experiments and analyses. Our experimental platform consists of an SPLIC environment with immersed servers, a suite of applications with various thermal footprints, and tools to monitor sensors and performance.

A. SPLIC and Server Hardware

SPLIC Environment: Our SPLIC environment is visualized in Figure 1. We use a Submer SmartPod v3, which utilizes a proprietary synthetic dielectric fluid to cool all computing hardware within the tank's volume. The pod uses two redundant pumps to circulate coolant. The pod exchanges heat with facility-chilled water, which flows through tubes at the bottom. The temperature of the in-flow chilled water is measured at 10 to 12 degrees Celsius.

We are limited to only a few servers available for immersion, which cannot generate enough heat in a short period for our studies if the pod is always connected to the chilled water. To address this limitation, we conduct thermal analyses while the chilled water is disconnected and continue after reconnecting chilled water to observe the system's capability to respond to strong cooling demand. This approach allows us to study the SPLIC system's response to high-heat environments without excessively stressing the system for prolonged periods. We maintain high levels of compute activity for an extended duration and then re-engage the chilled water, permitting the system to return to its normal state.

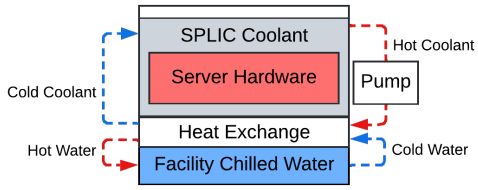

Fig. 1. SPLIC tank arrangement used in our experiments.

Server Hardware: We immerse three server nodes into the pod. The nodes are composed of commodity hardware and colloquially referred to as "deepgreen," "n01," and "n02", as detailed in Table I. The deepgreen node is the head node while n01 and n02 are compute nodes.

Hardware Preparation: The SPLIC coolant fluid includes plasticizer, so we replace thermal paste between hardware components and heat sinks with indium foil. The plasticizer also makes ethernet and power cables rigid and somewhat brittle after a few months. However, these cables remain at negligible risk of breaking while undisturbed.

We remove or disable all moving hardware on power supplies, motherboards, GPUs and CPUs. Server components physically moving in an air-cooling environment are unsuitable for immersion cooling. Take fans, for example; they would generate enormous heat from friction with the dense synthetic fluid, triggering alarms and risking burning[1]. Specialized hardware "immersion ready" or designed to improve SPLIC performance circumvents these issues. In this work, we are limited to commodity servers, which generalize across various possible server configurations.

TABLE I
Immersed Server Hardware

Nodes	Kind	Hardware	Specs
Deepgreen	Mother-board	S8021GM2NR-2T	Five (5) PCIe 3.0 x16 slots
	CPU	AMD EPYC 7551P	2.00 GHz 32 cores
	GPU Titan V	Two (2) NVIDIA 12 GB HBM2	5120 cores
n01, n02	Mother-board	X9DRG-QF	Four (4) PCIe 3.0 x16 slots
	CPU	Intel(R) Xeon(R) CPU E5-2670 v3	2.30 GHz 64 cores

B. Applications

We utilize a selection of HPC applications with different compute intensities, as shown in Table II.

TABLE II
Selected Applications and Classifications

Application Name	GPU Accelerated	Throughput Bottleneck
CUDA Stream	✓	Memory
EMOGI	✓	Memory
CUDA DGEMM	✓	Compute
MD5 Bruteforcer	✓	Compute
NPB DT Class=C	X	Memory
NPB IS Class=D	X	Memory
NPB EP Class=E	X	Compute
HPCC HPL	X	Compute

GPU Applications: We select CUDA Stream as a GPU-enabled memory bandwidth test with minimal computation. For a more realistic representation of memory-intensive GPU-accelerated HPC applications, we utilize a specially constructed graph using the EMOGI graph tool [5].

We also include compute-intensive GPU applications, including CUDA DGEMM which are widely used by many scientific and machine learning workloads. We also provide the MD5 Bruteforcing algorithm as an example of datacenter workloads. This applications' repeated hashes permit long arithmetic sequences without dependence upon large-scale memory accesses.

CPU Applications: We utilize several key kernels from the NAS Parallel Benchmark Suite [1], including **D**ata **T**raffic, **I**nteger **S**ort and **E**mbarassingly **P**arallel. Each of these benchmarks are executed at the largest class size that can be executed while utilizing the entire server system.

[1]We indeed experienced this once.

C. Metrics and Monitoring Tools

We utilize multiple publicly available software tools to collect metrics on CPU, GPU and NVMe device thermal sensors and activity measures, as well as a vendor-provided API to monitor the SPLIC system. To minimize the impact of monitoring on application performance, our tool is designed to sample each metric at a target rate of 1Hz and caches some information to reduce the overhead of sample collection. Our software harness is publicly available via GitHub[2].

CPU Monitoring: We record per-core frequencies from the `cpufreq/scaling_cur_freq` files in the Linux `/sys/devices/system/cpu/cpu*/` directories. These frequencies are recorded by the CPU governor in these files as integer-valued KHz.

We also use the libsensors library provided by lm-sensors (version 3.6.0) to monitor CPU core temperatures. The temperature values are reported as floating-point degrees Celsius with a single point of precision.

GPU Monitoring: We record GPU metrics listed in Table III using the NVML library based on NVIDIA driver version 535.54.03 and NVML version 12020, corresponding to CUDA version 12.2. Each metric is represented using integer values, so temperature data are reported less precisely for GPUs than equivalent measurements for CPU temperatures.

TABLE III
NVIDIA GPU METRICS RECORDED USING NVML

Metric	Meaning (Units)
GPU Temperature	Average SM temperature (°C)
Memory Temperature	Memory junction temperature (°C)
Power Usage	Power draw (mW)
Power Limit	Maximum power draw (mW)
GPU Utilization	NVIDIA metric of SM activity (%)
Memory Utilization	NVIDIA metric of memory activity (%)
Memory Used	Allocated global memory (bytes)
Performance State	NVIDIA metric of device state (integer 0-8)

NVMe Monitoring: We record NVMe temperatures using libnvme version 1.6, which are also reported as integer degrees Celsius.

PDU Monitoring: We record single-precision PDU phase amperage to determine the complete system's power draw over an SNMP interface. The Rack PDUs are Schneider Electric model AP7811B, with a 30 Amp limit and 208 Volt output of single-phase AC current, so the conversion to kiloWatts is straightforward.

Pod Monitoring: We use the provided API endpoint to collect metrics listed in Table IV using LibCurl version 7.68.0. The API reports temperatures at single-precision and most other values as integers.

D. Experimental Procedure

We conduct independent tests utilizing a single application as the workload for the server for the entire observed duration. Each test begins with thirty minutes of idling activity while disconnected from chilled water to establish a baseline for experimental conditions. After this initial period, we repeatedly execute the workload application for 7.5 hours, after which we reconnect the chilled water supply to the pod. We continue to monitor temperatures for an additional 24 hours as the pod dissipates accumulated heat and returns to its equilibrium state.

[2]https://github.com/tlranda/LibSensorsTools

TABLE IV
POD METRICS RECORDED FROM API

Metric	Meaning (Units)
Temperature	Average coolant temperature (°C)
Consumption	Pod power consumption (W)
Dissipation	Thermal dissipation (both as °C and KW)
mPUE	Power Usage Effectiveness (scalar)
Pump RPM	Pump rotations per minute (scalar)
Coolant Temperatures	Input/output coolant temperature (°C)
Water Temperatures	Input/output chilled water temperature (°C)
Flow Rates	Flow rate of coolant and water (L/minute)

V. RESULTS

We report our results based upon the research questions posed in Section III.

A. Thermal Behaviors of Individual Hardware Components (RQ1)

Importance. The range of measured temperatures of each hardware component determines the risk of violating thermal tolerances and how quickly temperatures may be expected to change. We also seek to identify variations in temperature based upon the application demand to determine if thermal behavior is purely driven by energy expenditure or if more complicated behaviors emerge.

Metrics. We begin by analyzing the minimum temperature as a small range of values collected during the idle baseline prior to each application test. As initial conditions are similar across all experiments, we aggregate all data during these periods to establish the minimum, mean and maximum observed temperatures of each component.

We then consider the recorded temperatures across each experiment's active application periods to determine the greatest attained temperature over the baseline and the greatest linear rate of temperature increase throughout all experiments. We compare the dissipated coolant heat relative to hardware activity measures to determine if more complex modeling is needed than pure power demand. The hardware activity measures and collection techniques are defined in Section IV-C.

Results. Each of the above metrics are computed for all monitored hardware and presented in Table V. All hardware components except the NVMe are capable of changing temperatures faster than the pod coolant, however only the GPU hardware was observed to reach levels where thermal throttling could become a reasonable concern when executing the DGEMM application, where it reached a maximum temperature of 96 °C. The Titan V GPU models are designed for a maximum operating temperature of 91 °C, with thresholds

for slowdown and shutdown at 97 and 100 °C, respectively. Had the experiment continued, the disconnected chilled water would not permit adequate heat exchange for the SPLIC system to protect immersed hardware and could have damaged the GPU. Notably, GPU temperatures fell to 46 °C almost immediately upon terminating the DGEMM application for this experiment, and a repeated execution of the benchmark without interrupted access to chilled water failed to exceed a GPU temperature of 80 °C. With the re-introduction of chilled water for heat exchange, the SPLIC system was able to return all temperatures to normal idling levels over the next 9 hours. Figures 2 and 3 show detailed temperatures of this experiment in without chilled water and with uninterrupted access to chilled water.

TABLE V
THERMAL BEHAVIORS BY HARDWARE COMPONENT

Hardware Component	Idle Temp. Min/Mean /Max (°C)	Interval (s) to Increase Temp by 1°C	Op. Temp. Observed Max (°C)
SPLIC Coolant	19.80 / 21.41 / 22.50	970.99	47.10
deepgreen CPU	14.75 / 17.51 / 19.66	702.20	55.62
deepgreen GPU	24.25 / 26.54 / 28.00	266.83	96.00
deepgreen NVMe	14.00 / 17.91 / 20.00	1068.48	44.00
n01 CPU	17.44 / 21.30 / 24.04	607.33	66.00
n02 CPU	17.67 / 20.81 / 24.26	560.35	69.00

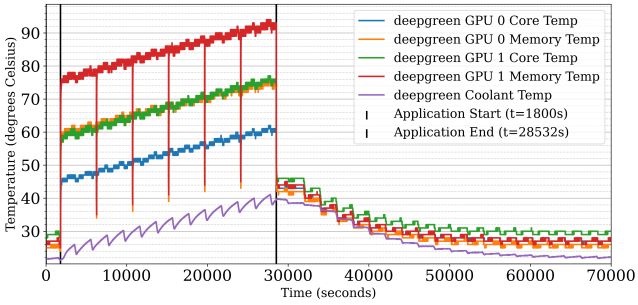

Fig. 2. Thermal Behavior of DGEMM Without Chilled Water. DGEMM runs on GPU 1.

B. Effects of Initial Conditions (RQ2)

Importance. SPLIC's energy efficiency may correlate with fluid temperature, making a trade-off between maximum thermal tolerance and long-term energy expenditure.

Metrics. We monitor the dissipated heat during periods where pumps are active and chilled water is available for the SPLIC heat exchange.

Results. We present the aggregated data across all observations in Figure 4. It's clear that the maximum heat removed positively correlates with a higher initial temperature,

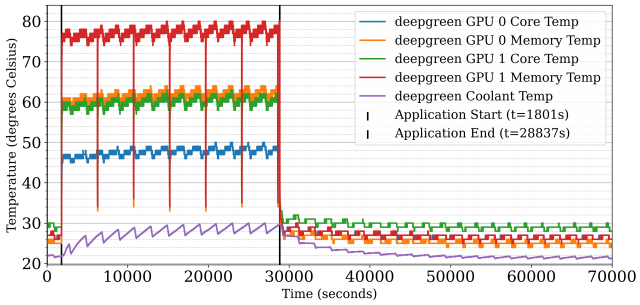

Fig. 3. Thermal Behavior of DGEMM With Chilled Water. DGEMM runs on GPU 1.

however the minimum heat removed during a cycle remains roughly constant regardless of initial temperature. Because the pump activity duration, chilled water temperature and pump power draw are held invariant in our experiments, this implies that SPLIC can yield increased energy efficiency at higher temperatures, however the behavior is not guaranteed to be observed.

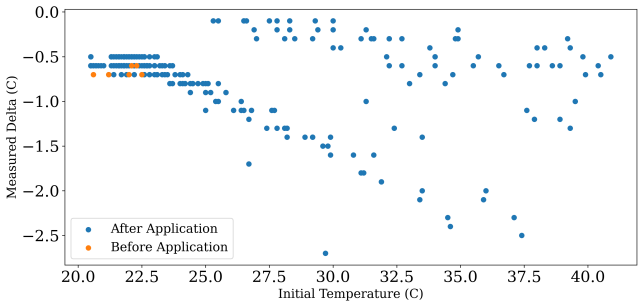

Fig. 4. Temperature at start of idle pump cycle vs amount of heat removed.

C. Effects of Workloads on Heat Accumulation (RQ3)

Importance. Given that almost all hardware components can heat up faster than the average coolant temperature, it's important to know what workloads will likely produce the greatest heat accumulation within the pod. This simplifies benchmarking a system's cooling demands and ensures system responses can be properly calibrated for extended application executions.

Metrics. We use the average interval to increase the temperature by one degree Celsius when the temperature delta reaches the maximum during application execution. To simplify comparisons, we group applications based upon CPU- or GPU-centric classifications and application throughput being Compute- or Memory-bound as previously denoted in Table II.

Results. We display the interval of each experiment application in Figure 5. The lower values indicate faster heat accumulation. These applications present a good range of heat accumulation rates. In contrast to common impression that GPU applications accumulate heat faster, only compute-intensive GPU applications do, while memory-bound GPU

applications yield relatively slow temperature changes. In general, compute-bound applications accumulate heat faster than memory bound applications, no matter whether they are programmed to run on CPU or GPU.

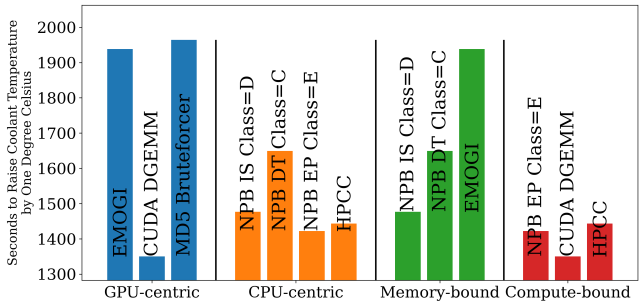

Fig. 5. Time intervals required by applications to accumulate heat. The smaller the interval, the faster the application execution increases coolant temperature by 1 °Celsius.

VI. Conclusions

Our experiments reveal a few initial insights about the current state of SPLIC immersion technology.

Strong resilience to thermal changes. Current tank designs use temperature readings at coolant pumps, instead of relying on embedded sensors in hotspots and computer components, to respond to thermal changes. This is a fixed strategy. As evidenced by our experiments, the technology can handle fluctuations in computing demand for extended durations. However, the same physical properties that enhance its energy efficiency also significantly extend the time required to return to a neutral state.

Delayed responses to temperature differentials. Our study shows that hot spots within hardware components have significant lag time before pod sensors can detect changes in temperature, meaning that responses must be calibrated to be more aggressive in inducing a cooling response in case hardware conditions outpace the system's detection. Future iterations of the hardware that can directly read computer components' sensors can respond in a more appropriate fashion without expending extra energy.

Ease of operation and maintenance. Modifying commodity hardware for initial server setup is intrusive but can be circumvented by utilizing specialty hardware designed for SPLIC. Non-specialized hardware can be limited by server layout, such as connectors that cannot bend sharply to reach ports and certain components that are inaccessible for maintenance without completely removing server blades due to the pod rack's vertical orientation. After removing hardware from the pod for maintenance, liquid residue can create slipping hazards and must be carefully monitored and cleaned. Finally, adding or removing components in the pod changes the level of coolant, which may require adding or removing coolant to maintain an appropriate volume for the container.

Reliability. Throughout our extended immersion of hardware, we have not observed degradation in peak performance. Plastic components affected by the plasticizer are eventually damaged and may require replacement during or after maintenance.

After our experiments, our datacenter had a power outage that we believe caused damage to a coolant pump sensor. This left the pod in a malfunctioning state that caused the coolant pumps to fail, leading us to terminate future experiments until a replacement could be sourced and installed. This level of operation disruption was not easily addressed by our technicians without vendor involvement, which complicates service and maintenance.

Acknowledgments

This research was partially supported by the U.S. National Science Foundation under Grant CNS-CCF-1942182.

References

[1] D. H. Bailey, *NAS Parallel Benchmarks*. Boston, MA: Springer US, 2011, pp. 1254–1259.

[2] R. Brown, A. to Save Energy, I. Incorporated, E. Incorporated, and U. E. P. Agency, "Report to congress on server and data center energy efficiency: Public law 109-431," Lawrence Berkeley National Laboratory, Tech. Rep., 2008, retrieved from https://escholarship.org/uc/item/74g2r0vg.

[3] *Computational Analysis for Thermal Optimization of Server for Single Phase Immersion Cooling*, ser. International Electronic Packaging Technical Conference and Exhibition, vol. ASME 2019 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems. Electronic and Photonic Packaging Division, 10 2019. [Online]. Available: https://doi.org/10.1115/IPACK2019-6587

[4] S. Liu, Z. Xu, Z. Wang, X. Li, H. Sun, X. Zhang, and H. Zhang, "Optimization and comprehensive evaluation of liquid cooling tank for single-phase immersion cooling data center," *Applied Thermal Engineering*, vol. 245, p. 122864, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359431124005325

[5] S. Min, V. S. Mailthody, Z. Qureshi, J. Xiong, E. Ebrahimi, and W. W. Hwu, "EMOGI: efficient memory-access for out-of-memory graph-traversal in gpus," *CoRR*, vol. abs/2006.06890, 2020.

[6] V. Natarajan, A. Deshpande, S. Solanki, and A. Chandrasekhar, "Thermal and power challenges in high performance computing systems," *Japanese Journal of Applied Physics*, vol. 48, no. 5S2, p. 05EA01, May 2009.

[7] N. A. Pambudi, A. Sarifudin, R. A. Firdaus, D. K. Ulfa, I. M. Gandidi, and R. Romadhon, "The immersion cooling technology: Current and future development in energy saving," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9509–9527, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016822001557

[8] J. M. Shah, K. Padmanaban, H. Singh, S. Duraisamy Asokan, S. Saini, and D. Agonafer, "Evaluating the Reliability of Passive Server Components for Single-Phase Immersion Cooling," *Journal of Electronic Packaging*, vol. 144, no. 2, p. 021109, 10 2021. [Online]. Available: https://doi.org/10.1115/1.4052536

[9] H. Shrigondekar, Y.-C. Lin, and C.-C. Wang, "Investigations on performance of single-phase immersion cooling system," *International Journal of Heat and Mass Transfer*, vol. 206, p. 123961, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0017931023001163