An Empirical Study of Thermal Attacks on Edge Platforms

Justin Duchatellier, Tyler Holmes
Kennesaw State University
Marietta, Georgia, USA
{jduchate,tholme60}@students.kennesaw.edu

Kun Suo
Kennesaw State University
Marietta, GA, USA
ksuo@kennesaw.edu

Yong Shi
Kennesaw State University
Marietta, GA, USA
yshi5@kennesaw.edu

ABSTRACT

Cloud-edge systems are vulnerable to thermal attacks as the increased energy consumption may remain undetected, while occurring alongside normal, CPU-intensive applications. The purpose of our research is to study thermal effects on modern edge systems. We also analyze how performance is affected from the increased heat and identify preventative measures. We speculate that due to the technology being a recent innovation, research on cloud-edge devices and thermal attacks is scarce. Other research focuses on server systems rather than edge platforms. In our paper, we use a Raspberry Pi 4 and a CPU-intensive application to represent thermal attacks on cloud-edge systems. We performed several experiments with the Raspberry Pi 4 and used stress-ng, a benchmarking tool available on Linux distributions, to simulate the attacks. The resulting effects displayed drastic increases in the temperature and power consumption. The key impact of our research is to highlight the following risks and mitigation plans: the vulnerability of cloud-edge systems from thermal attacks, the capability for the attacks to go unnoticed, to further the understanding of edge devices as well as the prevention of these attacks.

CCS CONCEPTS

• **Hardware** → *Thermal issues*; *Platform power issues*; • **Security and privacy** → Side-channel analysis and countermeasures; • **General and reference** → **Performance**.

KEYWORDS

Edge Computing, Thermal Side Channels, CPU Throttling

ACM Reference Format:
Justin Duchatellier, Tyler Holmes, Kun Suo, and Yong Shi. 2021. An Empirical Study of Thermal Attacks on Edge Platforms. In *2021 ACM Southeast Conference (ACMSE 2021), April 15–17, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3409334.3452071

1 INTRODUCTION

High performance computing devices enable rapid data processing in both cloud and edge-based Artificial Intelligence (AI) platforms. Powerful CPUs and GPUs are used in cloud and edge devices to execute compute-intensive processes. A byproduct of these processed data is heat, which will not only damage a physical system, but also lessen the system's performance. Generally, active and passive cooling systems are used to prevent the devices from overheating. However, if the temperature threshold is exceeded, the processor's maximum frequency will be throttled to reduce the amount of heat generated, which would significantly lower the performance of the processor and reduce the system's memory bandwidth [30]. Prior research has demonstrated that uncontrolled heat management can decrease both a system's performance and its usability. In public cloud and computing edge [31–33], where high performance and low latency are prime factors, excessive temperatures can dramatically undermine their availability.

In this paper, we performed several experiments on the representative device to highlight thermal threats on edge platforms. Thermal attacks were simulated using stress-ng [8], which is a stress test on Linux systems. In our first experiment, we monitored the temperatures of the edge device while the CPU was throttled using different maximum frequencies. Our second experiment compared the available cooling strategies for edge devices: no cooling, passive cooling (using heat sinks only), and active cooling (the fan's speed is either dynamically or statically controlled and paired with a heat sink). Our third experiment measured the power consumption and temperatures during thermal attacks on real-world edge AI workloads. We found that thermal effects are more pronounced when operational power is higher and maximum CPU frequency is higher. Various cooling strategies have significant impacts on application performance and energy consumption. Excessively high temperatures and long-term changes will greatly affect the quality of service and the life of hardware. In addition, as thermal attacks are similar to compute-intensive applications, it is difficult to accurately locate the source of overheating (the attacker). Complex

Table 1: Power Consumption of Servers and Edge Devices

Components	Power (Watts)
Intel Mid End CPU (Core i5)	73 ∼ 95
AMD Mid End CPU (4 cores)	65 ∼ 125
Regular Motherboard	25 ∼ 40
DDR3 RAM (1.5 Volts)	2 ∼ 3
Mid End Graphics Card	110 ∼ 164
2.5" Hard Disk Drive HDD	0.7 ∼ 3
120 mm Case Fan (2,000 RPM)	3.6 ∼ 6
Arduino Uno SoC	0.315 ∼ 0.96
Raspberry Pi 4B SoC	2.7 ∼ 6.4
NVIDIA Jetson TX2 SoC	7.5 ∼ 15

ACMSE 2021, April 15–17, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8068-3/21/04…$15.00
https://doi.org/10.1145/3409334.3452071

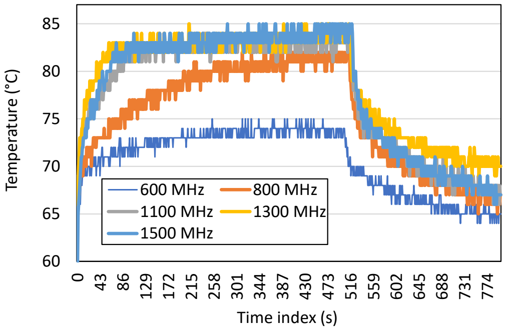

Figure 1: Raspberry Pi 4 CPU Throttled Temperatures

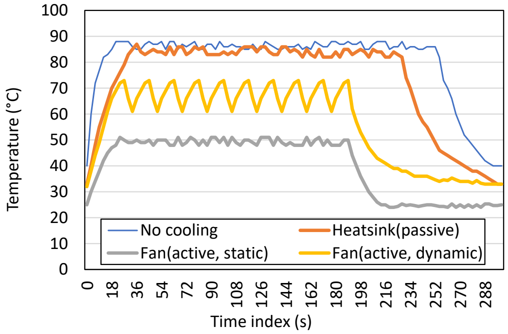

Figure 2: Cooling Strategies Comparison

cooling strategies and edge distributed power systems (such as batteries) make it more difficult to detect thermal attacks.

The following paper is organized as follows. A background of cooling systems is presented in Section 2 which summarizes the different cooling configurations in clouds and edges. Section 3 described our methodology, evaluation process, results, and analysis. Section 4 exhibits research from related works that cover thermal attacks and advances in energy saving systems. Section 5 summarize this paper with our insights and conclusion.

2 BACKGROUND OF COOLING SYSTEMS

As shown in Table 1, every component in cloud servers consumes lots of energy [6]. Most of the electrical power used for data center workloads is converted into heat as a byproduct. This heat must then be transported away from the equipment to prevent damages, or even fires, from occurring. The process requires an efficient heat removal method, an adequate air distribution type, and an appropriately positioned air cooling unit. Currently, there exist several air conditioning configurations for traditional data centers: rack, room, and row-based cooling [16]. Each configuration uses a computer room air handler (CRAH) that uses a chilled water valve to cool the supplied air for the intake. The name of the configuration refers to the location of the CRAH. Using a CRAH instead of a computer room air conditioner (CRAC) is a common practice because it is more cost-effective and easier to manage.

The amount of data, in both its existence and creation, has grown exponentially. The data center that undertakes the calculation and storage of these large amounts of data has also seen the increasing development of high density and huge power. The traditional air cooling system has gradually become overwhelmed under this trend. As a result, several companies have implemented liquid cooling. In the early 1960s, IBM began to explore water cooling for its mainframe computers [13]. Recently, Microsoft's Project Natick [5] team even deployed the Northern Isles data center under the sea in Scotland and tested the performance and reliability of its servers. However, due to safety concerns and high reconstruction costs, liquid cooling technology is still mainly limited to the field of high-performance computing (HPC).

Compared to cloud data centers, edge data centers have much lower power capacities (50-150 kW) [4], and they may use different cooling configurations. One of the most efficient cooling methods uses row-based cooling alongside a hot/cold aisle configuration. Row-based cooling houses the air cooling unit in between the server cabinets and directly next to the hardware. The server racks are configured so that hot air passes into a duct system and cool air passes through the servers' air intake. Each server would be facing the same way on the rack, in order to maximize the effectiveness of heat removal. Miniature-sized edge networks, consisting of one to five devices, that are deployed in an outdoor environment do not have air-conditioned cooling. For this reason, dynamic frequency fans or integrated liquid cooling are more commonly used. New architectures, software, and dynamically controlled energy saving components are being widely adopted for edge systems.

3 THERMAL THREAT MODEL AND RESULTS

In this section, we demonstrate the thermal threat on edge devices and describe our system's configuration. We also explain our methodology, present our preliminary results, and give our corresponding analysis.

3.1 Evaluation and Methodology

We conducted our study on representative edge hardware devices. In this research, we selected the Raspberry Pi 4B, which is equipped with a quad-core CPU, 4GB of RAM, and a 32GB SD card for storage. We also performed experiments on the Google Coral and NVIDIA Jetson TX2 and obtained similar observations. We used several different workloads in our experiments. To throttle the CPU and simulate a thermal attack, we executed *stress-ng* [8], a stress workload, on the Raspberry Pi. The options used in the test caused each CPU core to run every available CPU stress test while every available matrix stress method ran in parallel. The device's core temperature was measured by using the command *vcgencmd* [7]. The maximum CPU frequency was set using the *cpufreq* interface and an available frequency was set through the *scaling_max_freq* option [9]. We also selected AI Benchmark [1], which contains 46 computer vision tests and 14 benchmark sections such as object classification, facial recognition, and image deblurring. AI Benchmark simulated the machine learning (ML) and artificial intelligence (AI) workloads widely deployed on edges.

The experiment for the CPU throttling required the heat sinks and fan to be removed from the device. Initially, heat sinks were installed on the CPU, USB 3.0 controller, and SDRAM chip. The cooling system prevented the CPU from throttling. After the cooling components were removed, the CPU's maximum frequency was

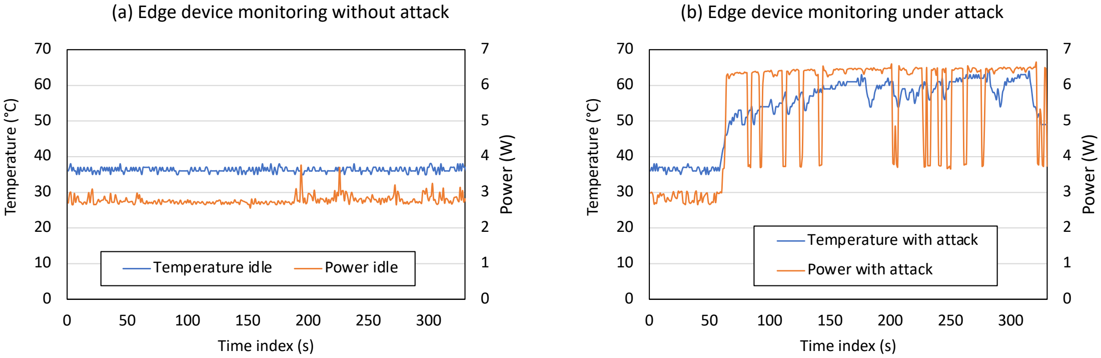

Figure 3: Power and Temperature Trace During Thermal Attack

reduced from 1500 MHz to lower frequencies: 1300MHz, 1100MHz, 800MHz, and 600MHz. Each iteration had a duration of 13 minutes and the stress test executed for seven minutes. The Raspberry Pi 4's cooling components were also removed while AI Benchmark was running.

The experiment to measure the temperature and power consumption was executed with only the fan installed on the device. We used a USB voltage tester to record the power consumption. This experiment executed for 10 minutes while the stress test lasted for 4 minutes. We also compared the device's performance with different cooling strategies: using a heat sink, statically controlled fans, and dynamically controlled fans.

3.2 Preliminary Results and Analysis

Figure 1 displays the throttled performance of the Raspberry Pi 4. As shown in Figure 1, the CPU's frequency was throttled the most at the highest frequency of 1500MHz. The CPU's frequency dropped approximately 7.14% at 1500MHz. The device required 200 seconds to return to its baseline of 64°C. In order to maximize the performance for edge devices, measures should be in place that actively monitor the CPU's current temperature. A list of these measures includes temperature sensors, limiting the maximum frequencies of processors, and systems that actively monitor power consumption. To evaluate the performance of real world applications under various CPU thermal effects, we further execute AI Benchmark on the Raspberry Pi 4. Since the AI Benchmark is a compute-intensive workload, the temperature of the CPU increases quickly and maintains a range between 79°C to 88°C during its execution. As shown in Figure 4, the CPU's temperature and the application's performance show a negative correlation. For instance, when the time index goes from 13000s to 16000s, the temperature rises slowly to a peak while the CPU's frequency slows down to about 600MHz. Similar results can also be observed during the time index from 22000s to 24000s. On the contrary, when the CPU's temperature gradually decreases to a low value, such as during 24000s to 26000s, the execution of the application will return to its maximum. When the CPU temperature is lower than 85°C, such as during the time index before 8000s or after 30000s, the application's speed is stable and maintains high performance.

Second, we measured the CPU's temperature and power consumption of edge devices. The Raspberry PI 4 consumed around 2.8W while idle and the processor's temperature was less than 40°C. As shown in Figure 3, during the stress test, the power consumption increased by more than 150% and the temperature nearly doubled. The effects from CPU throttling were still visible after our thermal attacks terminated. The device remained overheated and required over 240 seconds for the CPU's temperature to normalize after the stress test finished. A highly elevated temperature with long-lasting changes might greatly affect edge and IoT device stability. Also, the thermal effect and power consumption varied under different CPU speeds. We observed that the CPU's speed decreased the least, and the power consumption was lower when the maximum of CPU frequency was set to 600MHz. In addition, normal behaviors that occur with compute-intensive workloads (e.g., Figure 4(a)), such as machine learning and artificial intelligence applications, could consume power at a rate analogous to a thermal attack (e.g., Figure 3(b)). Malicious attackers could therefore camouflage themselves and attack a system with the user being unaware of the cause. To prevent such an occurrence, system and temperature logs should be kept, and abnormally behaving processes should be killed.

Third, we also studied how different radiators and cooling fans affect when thermal events happen, and compared their performance when running AI workload on edge devices. We deployed an image recognition application using an Inception V3 model on the Raspberry Pi. As depicted in Figure 2, without any cooling, the application's performance will be affected as the temperature of the processor reaches its maximum frequency. Passive cooling components such as heat sinks can slightly address the problem but cannot entirely prevent the issue. The application's performance will still be limited when the edge device's temperature reaches its threshold. Static, dynamic cooling configurations can effectively control the temperature and keep the application executing at the processor's full speed. However, it will waste more energy and increase the total power consumption. Lastly, the active and static cooling shows great performance improvement with over 30% execution time reduction compared to no cooling and 25% less energy cost compared to active, dynamic cooling. However, one downside of using an actively or passively cooled edge device is that detecting the location of the overheating source (e.g., attackers) may be difficult.

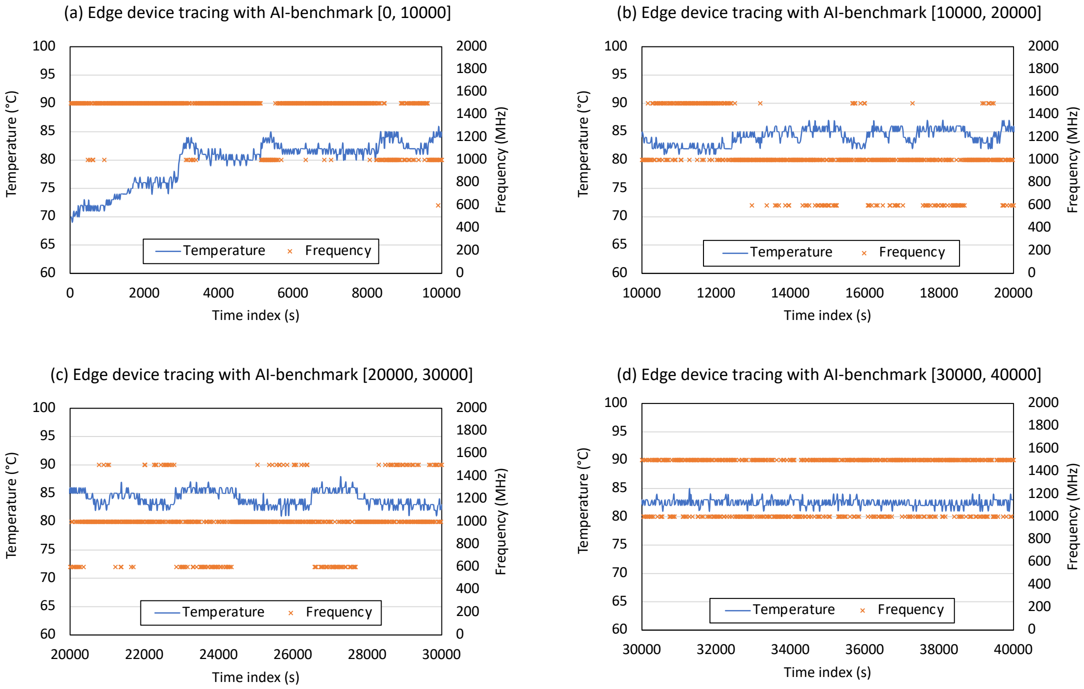

Figure 4: CPU Frequency and Temperature with AI Benchmark

4 RELATED WORK

Current research detailing thermal effects widely exists for cloud platforms. This section discusses these works and their limitations, as well as current energy efficient systems. In this research, we focus on limitations and inefficiencies that edge devices possess, and aim to reveal the emerging threat of thermal attacks on edge and IoT platforms.

4.1 Energy Efficient Systems

As edge systems are increasingly adopted, how to increase energy saving and efficiency in these systems captures more attention. To overcome the challenges of energy efficiency in edge systems, many hardware solutions including new IoT chips (i.e., ARM Cortex-M23 [2], Cortex-M33 [3]), new architectures (i.e., 3D chip design [25, 37]), and customized SoCs (e.g., AI-acceleration chips [12, 15, 36]) have been proposed. However, these solutions cannot work in the existing IoT or edge devices. Computation offloading (i.e., MAUI [14], ThinkAir [23]), a software solution, has been widely studied in recent years to alleviate heavy overhead and energy consumption at the edge. However, the efficiency of offloading highly depends on the network channel condition, as the implementation requires high data transmission. Recently, dynamic offloading [11, 20, 24, 26, 27, 29], which incorporates the characteristics of wireless channels, transmission power, computation load distribution, and heterogeneous types of computation tasks, with computation offloading algorithm was proposed for multi-user mobile cloud systems. Nevertheless, these works assume non-adjustable processing capabilities on the hardware, which is not energy efficient since the CPU's energy consumption at the edge increases exponentially with the processor's frequency [10].

Another notion of solution is dynamic voltage and frequency scaling (DVFS). There have been recent efforts to utilize DVFS to improve energy efficiency while enhancing scalability, manageability, and security [19]. To reduce system power consumption, an energy-aware scheduling algorithm, that uses DVFS, for parallel applications in heterogeneous distributed computing systems was developed [34], and a slacking algorithm for adjusting the CPU's frequency dynamically to extend a task's execution time was proposed [21]. Yet, these approaches did not perform well in handling communication-intensive applications or highly dynamic edge systems. In addition, most existing algorithms based on DVFS focus on shortening the scheduling time instead of optimizing the energy cost. In this work, we focus on investigating the relationship between energy saving techniques, processor processing, and heat control on the edge systems.

4.2 Thermal Attacks

Edge devices, as well as other computing devices, produce heat as a side effect from computation. The resulting heat traces could be exploited, and a user's credentials or information about the system could be exposed. Aside from the obvious security risks associated with leaked credentials, malicious actors could utilize heat traces to time thermal attacks on systems. The attacks could subsequently halt system operations, increase power consumption, send covert communications, and have other varying negative consequences. Kong et al. [22] found that malicious codes can exploit the deficiency and cause fine-grained, localized hotspots in the instruction cache, which might lead to physical damages. Szefer et al. [35]

discovered thermal channels can be used to create covert channels between users renting the same FPGA over time. Masti et al. [28] utilized thermal side channels in a multiprocessor system to send communications. They used PolarSSL to perform RSA decryption on select CPU cores and uncovered lesser known vulnerabilities of edge devices. Gao et al. [17] introduced the security concept of thermal attacks inside the data center that exploits thermal-intensive workloads to severely worsen the temperatures in the data center. To unveil the vulnerability of a data center to thermal attacks, they also conducted thermal measurements and proposed corresponding effective thermal attack vectors. In their following work [18], they further conducted tests on the thermal measurements of data centers from attacks through various scenarios. Damage assessment of these tests concluded that the attacks compromised server reliability and performance, increased cooling costs, caused cooling failures that can lead to server shutdowns, and caused local hotspots. Gao et al. proposed dynamic, thermal-aware load balancing to distribute workloads amongst servers based on thermal measurements and server location. This paper focuses on the emerging threat of thermal attacks in the edge system and these studies are orthogonal to our work.

5 CONCLUSION

In this paper, we presented the effects that temperature has on edge devices. Our experiments demonstrate how edge devices perform during throttling and thermal attacks. When throttled, the system required up to half of the time of the stress test's duration to return to its baseline temperature. This significant decrease in performance could be avoided by ensuring an active cooling system is implemented. The power consumption and temperature increased by nearly 105% and 70% respectively during the simulated thermal attack. A malicious actor could stealthily time such an attack to occur simultaneously with expected heavy workloads. Proper security policies and monitoring systems would prevent this from occurring. It is evident that edge devices have exploitable security vulnerabilities. Thermal attacks may go undetected if launched during CPU-intensive processes, which pose a threat to affected systems. Our research may be used for future hardware planning purposes, and may serve as guidance for best practices in hardware security at edge platforms or IoT devices.

REFERENCES

[1] [n.d.]. AI-Benchmark. http://ai-benchmark.com/.

[2] [n.d.]. Cortex M23. https://bit.ly/35Q5TJt.

[3] [n.d.]. Cortex M33. https://bit.ly/3lPN26E.

[4] [n.d.]. Edge Data Centers. https://bit.ly/3lSmv8X.

[5] [n.d.]. *Microsoft Finds Underwater Datacenters are Reliable, Practical and Use Energy Sustainably.* https://bit.ly/32ZRDvN.

[6] [n.d.]. *Power Consumption of PC Components in Watts.* https://bit.ly/3lM2aSB.

[7] [n.d.]. Raspberry Pi Documentation. https://bit.ly/3m4wlEV.

[8] [n.d.]. *Ubuntu Wiki: stress-ng.*

[9] D. Brodowski. [n.d.]. *CPU Frequency and Voltage Scaling Code in the Linux(TM) Kernel.*

[10] T. Burd and R. Brodersen. 1996. Processor Design for Portable Systems. *Journal VLSI Signal Processing Systems for Signal, Image and Video Technology* 13, 2-3 (1996), 203–221.

[11] X. Chen. 2014. Decentralized Computation Offloading Game for Mobile Cloud Computing. *In Proceedings of IEEE Transactions on Parallel and Distributed Systems (TPDS)* (2014).

[12] Y. Chen, T. Luo, S. Liu, S. Zhang, L. He, J. Wang, L. Li, T. Chen, Z. Xu, N. Sun, and O. Temam. 2014. DaDianNao: A Machine-learning Supercomputer. In *Proceedings of IEEE/ACM International Symposium on Microarchitecture (MICRO).* Cambridge, UK.

[13] J. Cortada. 2019. Building the System/360 Mainframe Nearly Destroyed IBM. *IEEE Spectrum* (April 2019).

[14] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. 2010. MAUI: Making Smartphones Last Longer with Code Offload. In *Proceedings of International Conference on Mobile Systems, Applications, and Services (MobiSys).* San Francisco, CA, USA.

[15] Z. Du, R. Fasthuber, T. Chen, P. Ienne, L. Li, T. Luo, X. Feng, Y. Chen, and O. Temam. 2015. ShiDianNao: Shifting Vision Processing Closer to the Sensor. In *Proceedings of International Symposium on Computer Architecture (ISCA).* Portland, OR, USA.

[16] K. Dunlap and N. Rasmussen. 2012. Choosing Between Room, Row, and Rack-based Cooling for Data Centers. https://bit.ly/2IZsuKJ.

[17] X. Gao, Z. Xu, H. Wang, L. Li, and X. Wang. 2017. Why "Some" Like It Hot Too: Thermal Attack on Data Centers. *SIGMETRICS Perform. Eval. Rev.* (2017).

[18] X. Gao, Z. Xu, H. Wang, L. Li, and X. Wang. 2018. Reduced Cooling Redundancy: A New Security Vulnerability in a Hot Data Center. In *Proceedings of Network and Distributed System Security Symposium (NDSS).* San Diego, CA, USA.

[19] J. Gong, J. Thompson, S. Zhou, and Z. Niu. 2014. Base Station Sleeping and Resource Allocation in Renewable Energy Powered Cellular Networks. *IEEE Transactions on Communications (TC)* (2014).

[20] D. Huang, P. Wang, and D. Niyato. 2012. A Dynamic Offloading Algorithm for Mobile Computing. *IEEE Transactions on Wireless Communications* 11, 6 (2012), 1991–1995. https://doi.org/10.1109/TWC.2012.041912.110912

[21] H. Kimura, M. Sato, Y. Hotta, T. Boku, and D. Takahashi. 2006. Emprical Study on Reducing Energy of Parallel Programs Using Slack Reclamation by DVFs in a Power-scalable High Performance Cluster. In *Proceedings of IEEE international conference on cluster computing (Cluster).* Barcelona, Spain.

[22] J. Kong, J. John, E. Chung, S. Chung, and J. Hu. 2010. On the Thermal Attack in Instruction Caches. *IEEE Transactions on Dependable and Secure Computing* (2010).

[23] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang. 2012. Thinkair: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading. In *Proceedings of IEEE Infocom.* Orlando, FL, USA.

[24] J. Kwak, Y. Kim, J. Lee, and S. Chong. 2015. DREAM: Dynamic Resource and Task Allocation for Energy Minimization in Mobile Cloud Systems. *IEEE Journal on Selected Areas in Communications* (2015).

[25] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir. 2006. Design and Management of 3D Chip Multiprocessors Using Network-in-memory. In *Proceedings of International Symposium on Computer Architecture (ISCA).* Boston, MA, USA.

[26] J. Liu, Y. Mao, J. Zhang, and K. Letaief. 2016. Delay-optimal Computation Task Scheduling for Mobile-edge Computing systems. In *Proceedings of IEEE International Symposium on Information Theory (ISIT).* Barcelona, Spain.

[27] Y. Mao, J. Zhang, and K. B. Letaief. 2016. Dynamic Computation Offloading for Mobile-edge Computing with Energy Harvesting Devices. *IEEE Journal on Selected Areas in Communications* (2016).

[28] R. Masti, D. Rai, A. Ranganathan, C. Müller, L. Thiele, and S. Capkun. 2015. Thermal Covert Channels on Multi-core Platforms. In *Proceedings of USENIX Security Symposium (USENIX Security).* Washington, D.C., USA.

[29] O. Munoz, A. Pascual-Iserte, and J. Vidal. 2014. Optimization of Radio and Computational Resources for Energy Efficiency in Latency-constrained Application Offloading. *IEEE Transactions on Vehicular Technology* (2014).

[30] R. Schöne, D. Hackenberg, and D. Molka. 2012. Memory Performance at Reduced CPU Clock Speeds: An Analysis of Current x86_64 Processors. In *Proceedings of Workshop on Power-aware Computing Systems (HotPower).* Berkeley, CA, USA.

[31] K. Suo, Y. Zhao, W. Chen, and J. Rao. 2018. An Analysis and Empirical Study of Container Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM).* Honolulu, HI, USA.

[32] K. Suo, Y. Zhao, W. Chen, and J. Rao. 2018. vNetTracer: Efficient and Programmable Packet Tracing in Virtualized Networks. In *Proceedings of International Conference on Distributed Computing Systems (ICDCS).* Vienna, Austria.

[33] K. Suo, Y. Zhao, J. Rao, L. Cheng, X. Zhou, and F. Lau. 2017. Preserving I/O Prioritization in Virtualized OSes. In *Proceedings of the 2017 Symposium on Cloud Computing (SoCC).* Santa Clara, California, USA.

[34] Z. Tang, L. Qi, Z. Cheng, K. Li, S. Khan, and K. Li. 2016. An Energy-efficient Task Scheduling Algorithm in DVFS-enabled Cloud Environment. *Journal of Grid Computing* (2016).

[35] S. Tian and J. Szefer. 2019. Temporal Thermal Covert Channels in Cloud FPGAs. In *Proceedings of ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA).* Virtual Event, USA.

[36] S. Zhang, Z. Du, L. Zhang, H. Lan, S. Liu, L. Li, Q. Guo, T. Chen, and Y. Chen. 2016. Cambricon-x: An Accelerator for Sparse Neural Networks. In *Proceedings of Annual IEEE/ACM International Symposium on Microarchitecture (MICRO).* Taipei, Taiwan.

[37] P. Zhou, P. Yuh, and S. Sapatnekar. 2010. Application-specific 3D Network-on-chip Design Using Simulated Allocation. In *Proceedings of 15th Asia and South Pacific Design Automation Conference (ASP-DAC).* Taipei, Taiwan.