Measuring and Exploiting Guardbands of Server-Grade ARMv8 CPU Cores and DRAMs

Measuring and Exploiting Guardbands of Server-Grade ARMv8 CPU Cores and DRAMs

Konstantinos Tovletoglou
Lev Mukhanov
Georgios Karakonstantis
Queen's University Belfast

Athanasios Chatzidimitriou
George Papadimitriou
Manolis Kaliorakis
Dimitris Gizopoulos
University of Athens

Zacharias Hadjilambrou
Yiannakis Sazeides
University of Cyprus

Alejandro Lampropulos
WorldSensing

Shidhartha Das
ARM Ltd.

Phong Vo
A.M.C.C. Deutschland AMPERE

Abstract - In this paper, we present the results of our comprehensive measurement study of the timing and voltage guardbands in memories and cores of a commodity ARMv8 based micro-server. Using various synthetic micro-benchmarks, we reveal how the adopted voltage margins vary among the 8 cores of the CPU chip, and among 3 different sigma chips and we show how prone they are to worst-case voltage noise. In addition, we characterize the variation of 'weak' DRAM cells in terms of their retention time across 72 DRAM chips and evaluate the error mitigation efficacy of the available error-correcting codes in case of operation under aggressively relaxed refresh periods. Finally, we show the overall energy savings that could be achieved by shaving the adopted guardbands in the cores and memories using various applications. Our characterization results show the potential to obtain up-to 38.8% energy savings in cores and up-to 27.3% within DRAMs.

I. Introduction

As transistors are being pushed to the atomic scale, it is becoming very difficult to fabricate circuits with the expected power and performance specifications leading to large static and dynamic variations [1]. To cope with the significant hardware variability and avoid the risk of system failures manufacturers try to hide it from the system software by adopting pessimistic voltage and frequency margins/guardbands based on the few worst-case manufactured chips and assumed scenarios that are rare to occur [1, 3-7]. Such guardbands end-up forcing the circuits to work less efficiently than they could, essentially increasing the power consumption and constraining performance of all the manufactured circuits based on the worst-case parts. Such margins are becoming more prominent with the use of more cores per chip and technology scaling. These technology trends exacerbate core-to-core variations, voltage droops [2, 3], reliability issues [2] and SRAM malfunctions [15,16] at low voltages (Vmin).

In this paper, we perform a characterization study of a commodity server, X-Gene2, equipped with 8 64-bit ARMv8 cores and 32GB DDR3 DRAM and a Linux Operating System. Such a server is a typical architecture of the latest generation of micro-servers that aim at improving the energy efficiency in cloud and edge data-centers. We measure the guradbands in 8 cores of ARMv8 CPU chips, and 3 different sigma chips manufactured in 28nm process based on an automated characterization methodology and by using various synthetic benchmarks and applications. To measure voltage guardbands, it is essential to craft worst-case voltage noise stress tests (also known as dI/dt viruses) that expose worst-case voltage droops [2,8,13]. Typically, these stress-tests are automatically generated using optimization approaches, such as Genetic Algorithms (GA), guided by direct voltage measurements [2,8]. Since X-Gene2 doesn't support fine-grained voltage measurements, to generate dI/dt viruses, we use an alternative methodology which is based on sensing high voltage noise through CPU electromagnetic emanations (EM) [14]. Essentially, we use GA to craft a loop of instructions that maximizes radiated EM amplitude. By maximizing EM amplitude, voltage noise is maximized as well, which we prove with Vmin testing. The dI/dt viruses are particular useful for exposing inter-chip process variations as different chips have different tolerance in voltage drops.

As both the CPU pipeline and cache memories operate under the same voltage domain, we can identify whether the chip failures rise from the cache memories or from pipeline logic by crafting synthetic programs that specifically target components in both regions [17]. These synthetic programs were developed to isolate particular components inside the CPU, including both L1 instruction and data cache memories, L2 cache as well as integer and FP ALUs. This is achieved by exploiting architectural and micro-architectural characteristics of the X-Gene2 platform and ARMv8 ISA. Workload variations are also present as different workload cause different voltage noise [2,3,8,14]. To capture these workload variations, we perform a comprehensive characterization using real workloads, including SPEC2006 and NAS benchmark suites in both single-process and multi-process setups. Finally, we also measure the retention time variation within 72 DRAM chips under various temperatures using a unique thermal testbed attached on the server board. For such a characterization, we used synthetic benchmarks based on worst-case data patterns as well various high performance computing (HPC) workloads. Our results indicate that there are extensive guardbands within cores and memories which if utilized can lead to up-to 38.8% energy savings in cores and up-to 27.3% within DRAMs. The characterization results could help guide the operation of the underlying hardware components within 'safe' operating points, which do not lead to system disruptions. Finally, once such 'safe' points were used for the execution of a novel multi-threaded denial-of-service attack detection application, we find that the server power could be reduced by 20.2% without any disruption.

The rest of the paper is organized as follows. Section II presents the architecture and circuit details of the server. Section III presents the characterization methodology. Section IV discusses the characterization results and power savings that can be achieved. Finally, conclusions are drawn in Section V.

II. Server Architecture Details

In this paper, we focus on the X-Gene2 Server-on-a-Chip (SoC), which is the latest generation of the X-Gene family of chips used in the popular HP Moonshoot servers [17]. As depicted on Figure 1, the X-Gene2 SoC consists of four processor modules (PMDs), each with two 64-bit ARMv8 cores running at 2.4GHz. The implemented memory hierarchy is representative of any modern high performance system consisting of a 32 KB L1 data cache and a 32 KB L1 instruction cache per core, a private 256 KB L2 cache shared between the two cores of each PMD and an 8 MB L3 cache shared across all cores through the cache-coherent Central Switch (CSW). The X-Gene2 has two Memory Controller Bridges (MCBs) which are connected to the CSW providing access to DRAM. In turn, each MCB is connected to two DDR3 Memory Control Units (MCUs). Each MCU has one channel of DDR3 memory and support up to two DIMMs with two ranks each.

The X-Gene2 provides access to a separate Scalable Lightweight Intelligent Management Processor (SLIMpro), a special management core, which is used to boot the system and provide access to on-board sensors for measuring the temperature and power of the SOC and DRAM. The SLIMpro also reports to the Linux kernel all errors corrected or detected by the provided error-correcting codes (ECC)

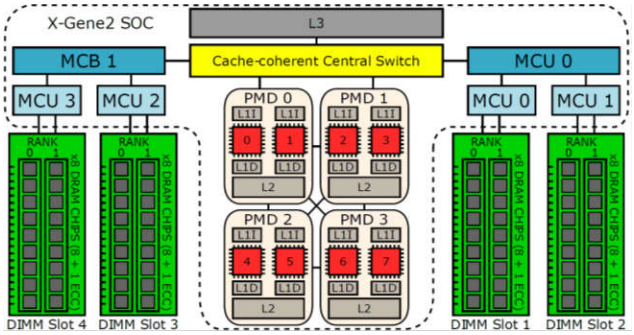


Fig. 1: X-Gene2 Server-On-Chip Architecture

and the parity. Finally, SLIMpro allows to configure the parameters of the MCUs, such as timings and the refresh period (T_{REFP}). The server runs a fully-fledged OS based on CentOS 7 with the default Linux kernel 4.3.0 for ARMv8 and support for 64KB pages.

III. Characterization Methodology

Heterogeneity exists among cores located on the same chip, DRAM and cache memory banks. Each resource may perform better or worse than others but certainly not as any other similar resource on the board. Therefore, there is a need to characterize each core and memory bank individually. To this end, we developed an automated characterization framework, as shown in Figure 2, (1) to identify the target system's limits when it operates at scaled voltage and frequency conditions, and (2) to log the effects of a program's execution under these conditions.

As shown in Figure 2, the characterization framework consists of three phases: initialization, execution, and parsing. During the initialization phase, a user can declare a benchmark list with corresponding input datasets to run in any desirable characterization setup. The characterization setup includes the voltage and frequency (V/F) values on which the experiment will take place and the cores where the benchmark will be run. The execution phase consists of multiple runs of the same benchmark, each one representing the execution of the benchmark in a pre-defined characterization setup. The set of all the characterization runs running the same benchmark with different setups represents a campaign. In the parsing phase of our framework, all log files that are stored during the execution phase are parsed in order to provide a fine-grained classification of the effects observed for each characterization run.

We have extended the error reporting capabilities of existing mechanisms (i.e. ECC in caches and DRAMs) with system configuration values, sensor readings and performance counters for identifying correctable (CE) and uncorrectable errors (UE). In addition, to account for any undetected error and essentially measure any Silent Data Corruption (SDC) that could go undetected by ECC, we compare the output of each execution with a golden reference.

A. Socketed Board for Sigma Chip Characterization

Our methodology has included normal chips as well as corner parts (*sigma chips*) in order to better expose the variation among different chips. For this purpose, special socketed validation boards were developed that allow the usage of typically discarded chips. The same boards were used for both normal (namely TTT) and sigma parts as a common base system to limit the system differences on chip level. Sigma chips were selected from both ends, which means that they were identified to have high or low leakage, both beyond nominal thresholds. The high leakage corner parts (namely TFF) can operate in higher frequencies, while the low leakage parts (TSS) in lower frequency. This also translates to higher and lower guardbands respectively.

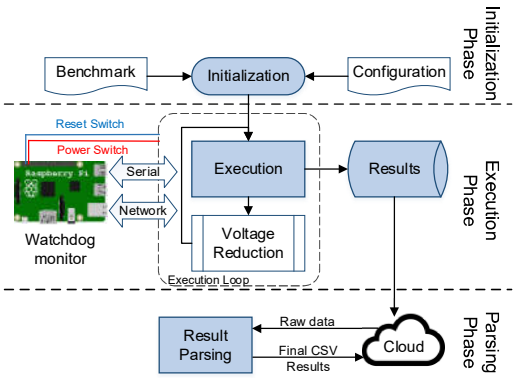


Fig. 2: Characterization framework layout.

B. Thermal Testbed for DRAM Characterization

Note that since temperature plays a significant role in the DRAM behavior, we also developed a first of its kind temperature-controlled testbed for DRAMs on a server. The testbed is based on adapters with heating elements and shown in Figure 3a. Each adapter consists of a resistive element, thermally conductive tape transferring the heat of the element to all the chips of a DIMM in uniform way and a thermocouple to measure the temperature. The temperature of each element is controlled by a controller board, as shown in Figure 3b, which contains a Raspberry Pi 3, four closed-loop PID controllers and eight solid state relays controlling the resistive elements of each DIMM and rank independently. By measuring the temperature on the DIMMs with both the thermocouple and the embedded sensor on the SPD chip, the controllers can aggressively control the heating elements and regulate the temperature. During our experiments, the maximum deviation from the set temperature is less than 1°C.

C. Stress-test development

Cores/Caches. To characterize the hardware components, we stress the underlying cores and memories using diagnostic viruses. Cache viruses were crafted to exploit the underlying microarchitecture and test all levels of the cache hierarchy, while core viruses are being generated by genetic algorithms. To stress the cores, we use dI/dt viruses that cause the CPU power consumption to switch between high and low power at a rate equal to PDN 1st order resonant frequency [2,8,14]. This causes maximum voltage noise. We craft this virus using the approach described in [14]. Such viruses represent a pathogenic worst-case scenario that is unlikely to be encountered in real-life workloads targeting to cause maximum voltage noise, power consumption and error rates. Despite the unlikeliness of these worst-case scenarios, the nominal operating voltages are still more pessimistic which limits the energy-efficiency of many chips that could operate with lower guardbands. This is mostly due to the fact that manufacturers have to account for process variations across different chips of the same model. Therefore, these worst-case stress test are useful for exposing hardware heterogeneity.

DRAMs. To characterize the DRAMs, we used data pattern benchmarks (DPBenchs) based on all 0s, all 1s, checkerboard and random data patterns which stress the whole DRAM memory by writing the specific patterns and accessing them. DPBenches were shown to be effective in stressing the DRAM cells and their retention time [19].

IV. Results and Saving Projections

In this section, we present our pre-deployment characterization results obtained in the initial phase of the project for the on-board cores and DRAMs within our ARMv8 server prototype for a variety of benchmarks. Furthermore, we analyze the potential power savings.

A. Characterization of CPUs

We experimentally obtain the Vmin values of the 10 SPEC CPU2006 [10] benchmark on the three X-Gene 2 chips (TTT, TFF, TSS) [11], running the entire time-consuming undervolting

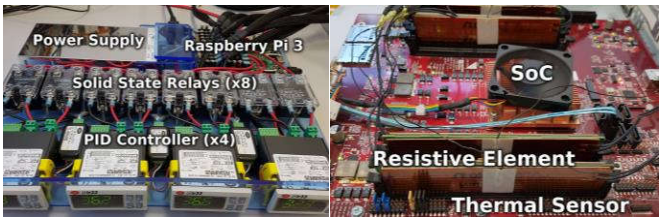


Fig. 3: a) X-Gene 2 with the thermal adapters, b) Temperature controller board.

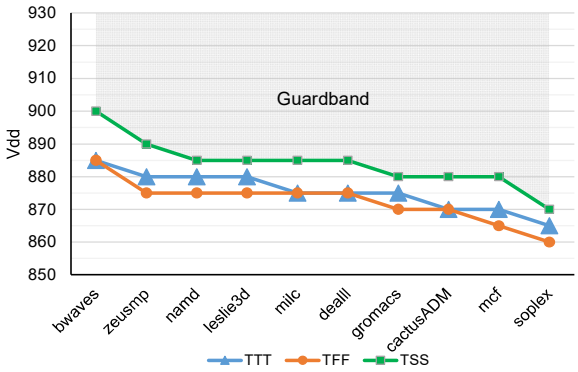


Fig. 4: Vmin results at 2.4 GHz for 10 SPEC2006 programs on 3 different X-Gene 2 chips (TTT, TFF, TSS)

experiment ten times for each benchmark, following the flow described in Section 3.A. This part of our study focuses on a quantitative analysis of the safe Vmin for diverse chips of the same architecture in order to expose the potential guardbands of each chip, as well as to quantify how the program behavior affects the guardband and to measure the core-to-core and chip-to-chip variation. For a significant number of benchmarks, we can see variations between different programs and different chips. Figure 4 represents the most robust core for each chip, and for these programs the Vmin varies from 885mV to 860mV for TTT, from 885mV to 870mV for TFF and from 900mV to 870mV for TSS. Considering that the nominal voltage for the X-Gene 2 is 980mV, there is a significant reduction of voltage without affecting the correct execution of programs, which is equal to at least 18.4% for the TTT and TFF chip, and 15.7% for the TSS chip. We also notice in Figure 4 that the workload-to-workload variation follows similar trends across the 3 chips of the same architecture; however, there is a relatively large variation among the chips. This means that there is a program dependency of Vmin behavior in all chips. Figure 5 shows the potential savings for the case that 8 different benchmarks run simultaneously: bwaves, cactusADM, dealII, gromacs, leslie3D, mcf, milc, namd. By exploiting the predictor's results, 12.8% power savings can be obtained by adjusting the voltage to the TTT Vmin without performance loss. Alternatively, the frequencies of the 2 weakest PMDs (0 and 1) can be reduced to 1.2 GHz (resulting in 25% performance loss) which will allow further reduction of the supply voltage to 885mV and energy savings up to 38.8%. Therefore, the predictor, apart from predicting the safe Vmin, can also assist task scheduling in conjunction to frequency scaling according to the current workload on the system to further improve energy efficiency.

B. Worst Case Voltage Noise Characterization and Exposing Inter-Chip Process Variation

Since X-Gene2 does not support fine-grained voltage measurements, voltage noise viruses are crafted using an alternative methodology which is based on sensing high voltage noise through CPU electromagnetic emanations (EM) [14]. To prove the effectiveness of the crafted virus, we use an indirect measurement of voltage noise which is Vmin (minimum operational voltage for a given frequency) [2,8,14]. In Figure 6, we observe that the EM virus has the highest Vmin compared to conventional workloads like NAS.

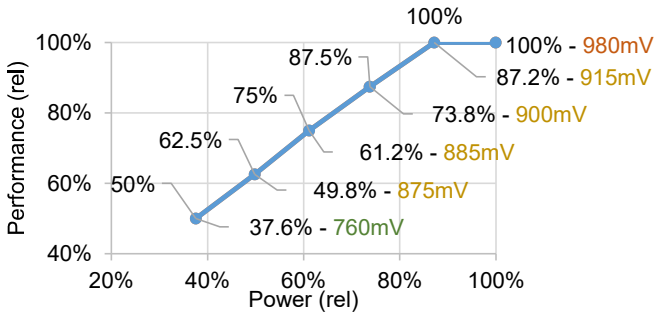


Fig. 5: Power/performance tradeoffs for an 8-benchmark workload

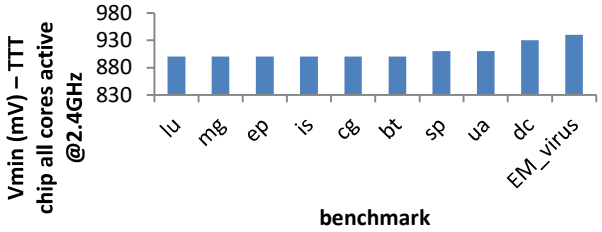


Fig. 6: Vmin of EM virus vs NAS benchmarks

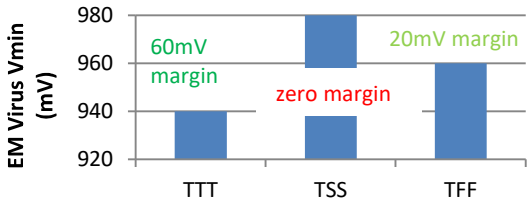


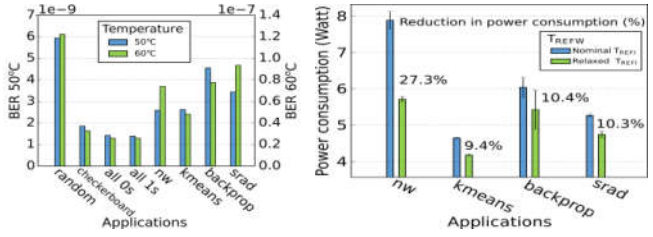
Fig. 7: Exposing inter-chip process variations with EM virus

Moreover, the EM virus allow us to expose inter-chip process variations. In Figure 7, we observe that TTT chip has 60mV margin. This implies that we can shave at least 50mV from this chip's operating voltage and improve the energy efficiency. On the other hand, the TSS chip doesn't seem to have any voltage margin as the virus crashes the system just 10mv below the nominal. Thereby, the TSS chip is better to be operated at manufacturer suggested nominal voltage.

C. DRAM Characterization

In our experiments, we characterize 72 DRAM chips [18] operating at 50 °C and 60 °C, under 35x relaxed refresh period, from the nominal 64ms to 2.283s, using the mentioned DPBenches.
Our results, revealed that i) all manifested errors are corrected by ECC and ii) there is large variation of the number of weak cells across the DRAM chips as depicted in Table I. In fact, we observed that the number of error-prone locations may differ by 41% within each chip from bank to bank at 50 °C and by 16% at 60 °C.

To explore a workload to workload variation of memory errors, we run four memory intensive HPC applications from the Rodinia benchmark suite (backprop, kmeans, nw and srad) and evaluated how the BER (the Bit-error rate) varies across benchmarks as depicted in Figure 8a. We observe that the BER varies by up-to 2.5 times. Our experiments revealed that the real workloads incur less BER than the virus based on random DPBench. This can be attributed to the fact that the data patterns actually stored in DRAM during the execution of real applications, may differ from the worst-case data patterns used in the synthetic benchmarks. Moreover, the HPC applications may also access the DRAM rows with a high frequency which is sufficient for inherently refreshing them and thus avoiding errors. As a side note, our results confirmed earlier studies [19] that the highest BER is observed for the random DPBench which implies that this DPBench could be used as a representative benchmark for characterization of DRAM error behavior.



Fig, 8: a) BER for DPBenches and Rodinia, b) Power savings for relaxed refresh period

TABLE I: Variation of the number of unique error locations across DRAM banks under different temperatures.

	1	2	3	4	5	6	7	8
50 ℃	180	213	228	230	163	198	204	208
60 ℃	3358	3610	3641	3842	3293	3448	3601	3540

Overall, we observed that the available SECDED ECC can detect and correct all manifested errors and avoiding any disruption of such a complete system, when the DRAM temperature does not exceed 60 °C. Figure 8b demonstrates the power savings achieved by relaxing the refresh period by 35x. We see that the power savings also vary and the maximum power gain is achieved for the nw (Needleman-Wunsch) benchmark at 27.3%, while the lowest is achieved by kmeans at 9.4%.

To exploit the findings of our characterization and the unique characteristics of each workload, we have attempted to reorder the issued memory accesses by ensuring that all accesses occur within a targeted time period that is less than the next scheduled refresh operation. Once we applied such a method to the popular Stencil algorithms, we observed that access intervals are shorter than the refresh period[12], indicating that such technique can be further exploited for limiting the manifested DRAM errors and reduce the reliance on ECC and required error corrections.

D. Exploitation of the Revealed Margins

The main aim of the characterization process is to reveal the 'safe' operating points in cores and DRAMs within each server and exploit them during system operation for saving energy without degrading the system availability/reliability. By doing so we can essentially trim the pessimistic guardbands adopted by manufacturers and utilize the true capabilities of each core and memory.

To showcase the possible savings at the overall server, apart from the CPU and memory intensive applications used above, we have executed a novel end-to-end Jammer detector application that aims at detecting devices that may cause Denial-of-Service attacks in wireless networks. Such an application is a real workload that will need to be executed in future edge environments, where all the communication of the Internet-of-Things rely on the availability of wireless networks. The developed Jammer detector application monitors with Software Defined Radio modules, the whole wireless spectrum to detect any anomalies and any device that may cause potential Denial-of-Service attacks. We are executing 4 parallel instances of the Jammer, in order to utilize the maximum CPU and memory bandwidth, while respecting the requirements of Quality-of-Service (QoS) and required response time of the detector.

As we discussed, based on our analysis above we have identified that TTT cores can run at 930mV (the PMD domain) and 920mV (the SoC domain) without causing any disruption, while DRAMs can operate with 35x relaxed refresh period. Using such new operating points during the execution of the Jammer application, our results show that we can reduce the total server power from 31.1W down to 24.8W and achieve 20.2% total power savings (Figure 9), without compromising the QoS constraints required by the Jammer. The energy savings are high in case of the PMD and DRAM domains (i.e. 20.3% and 33.3% correspondingly), while the SoC domain results in 6.9% savings.

The characterization results could finally be used to develop a module for predicting the hardware behavior and suggesting optimistic 'safe' operating points to the Linux governor, which is the future aim of our work. To achieve this, we can train a workload dependent prediction model considering also performance counters as we recently proposed in [11]. Such a model can take also into consideration the history of voltage droops occurred over time. Then based on a chip's intrinsic Vmin (this can be determined with idle Vmin test) and the history of droops, we can predict the probability of the operating voltage crossing the intrinsic Vmin. This leads to predicting the probability of failure at various operating voltages. Solid prediction will help establishing a robust and efficient online voltage adoption mechanism.

Fig. 9: Power consumption per domain for nominal and undervolted operation

V. CONCLUSIONS

This paper presents a comprehensive analysis of the guardbands adopted within memories and cores of a commodity ARMv8 based micro-server. Our results show that there are extensive pessimistic margins that have been adopted in the supply voltage of the cores and the refresh rate of the DRAMs. By trimming such guardbands and allowing operation at the 'safe' operating points according to the capabilities of the cores and memories then the energy of the server could be reduced by 20.2%, without disrupting the system operation.

ACKNOWLEDGEMENTS

The work presented was partially supported by the UniServer project under the EU Horizon 2020 programme and grant no. 688540.

REFERENCES

[1] K. A. Bowman, et al. "A 45 nm resilient microprocessor core for dynamic variation tolerance". IEEE JSSC 2011.

[2] P. N. Whatmough, et al. "14.6 an all-digital power-delivery monitor for analysis of a 28nm dual-core arm cortex-a57 cluster." ISSCC 2015.

[3] V. J. Reddi et al. "Voltage smoothing: Characterizing and mitigating voltage noise in production processors via software-guided thread scheduling." MICRO 2010.

[4] S. Borkar et al. "Parameter variations and impact on circuits and microarchitecture." DAC, 2003.

[5] H. Esmaeilzadeh et al. "Dark silicon and the end of multicore scaling." ISCA 2011.

[6] G. Karakonstantis et al. "Containing the nanometer pandora-box: Cross-layer design techniques for variation aware low power systems." IEEE JETCAS 2011.

[7] L. Leem et al. "Cross-layer error resilience for robust systems." IEEE ICCAD 2010.

[8] Y. Kim, et al. "AUDIT: Stress testing the automatic way." MICRO 2012.

[9] OpenStack. "Open source software for creating private and public clouds." [Online]. Available: https://www.openstack.org/.

[10] J. L. Henning. "SPEC CPU2006 benchmark descriptions." SIGARCH Comput. Archit. September 2006.

[11] G. Papadimitriou et al. "Harnessing voltage margins for energy efficiency in multicore CPUs." MICRO 2017

[12] K. Tovletoglou et al. "Relaxing DRAM refresh rate through access pattern scheduling: A case study on stencil-based algorithms." IOLTS 2017

[13] Bertran, Ramon, et al. "Voltage noise in multi-core processors: Empirical characterization and optimization opportunities." MICRO 2014.

[14] Hadjilambrou, Zacharias, et al. "Sensing CPU voltage noise through Electromagnetic Emanations." IEEE CAL 2017.

[15] S. Ganapathy, et al. "On Characterizing Near-Threshold SRAM Failures in FinFET Technology." DAC 2017.

[16] C. Wilkerson, et al. "Trading off cache capacity for reliability to enable low voltage operation." ISCA 2008.

[17] G. Singh et al. "AppliedMicro X-Gene2." Hot Chips 2014.

[18] Micron. "DDR3 SDRAM MT41J512M8." 2009

[19] J. Liu et al. "An experimental study of data retention behavior in modern dram devices: Implications for retention time profiling mechanisms." ISCA 2013.

[20] G. Papadimitriou et al. "Micro-Viruses for Fast System-Level Voltage Margins Characterization in Multicore CPUs." ISPASS 2018.