**QUEEN'S
UNIVERSITY
BELFAST**

Characterization of HPC workloads on an ARMv8 based server under relaxed DRAM refresh and thermal stress

Mukhanov, L., Tovletoglou, K., Nikolopoulos, D., & Karakonstantis, G. (2018). Characterization of HPC workloads on an ARMv8 based server under relaxed DRAM refresh and thermal stress. In *Proceeding SAMOS '18 : Proceedings of the 2018 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS), Greece. 2018* (pp. 230-235). ACM.
<https://doi.org/10.1145/3229631.3236091>

Published in:
Proceeding SAMOS '18

Document Version:
Peer reviewed version

Queen's University Belfast - Research Portal:
[Link to publication record in Queen's University Belfast Research Portal](#)

Publisher rights
Copyright 2018 IEEE. This work is made available online in accordance with the publisher's policies. Please refer to any applicable terms of use of the publisher.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Characterization of HPC workloads on an ARMv8 based server under relaxed DRAM refresh and thermal stress

Lev Mukhanov

ECIT, Queen's University Belfast, School of EECS
l.mukhanov@qub.ac.uk

Dimitrios S. Nikolopoulos

ECIT, Queen's University Belfast, School of EECS
d.nikolopoulos@qub.ac.uk

Konstantinos Tovletoglou

ECIT, Queen's University Belfast, School of EECS
k.tovletoglou01@qub.ac.uk

Georgios Karakonstantis

ECIT, Queen's University Belfast, School of EECS
g.karakonstantis@qub.ac.uk

ABSTRACT

Improving energy efficiency of the memory subsystem becomes increasingly important for all digital systems due to the rapid growth of data. Many recent schemes have attempted to reduce the DRAM power by relaxing the refresh rate, which may negatively affect the DRAM reliability. To optimize the trade-offs between power and reliability, existing studies rely on experimental setups based on FPGAs and the use of few known data-patterns for exciting rare worst-case circuit reliability effects. However, by doing so, existing studies may be missing to capture the real DRAM behavior within a commodity server with a fully fledged OS.

In this paper, we develop an experimental framework based on a state-of-the-art 64-bit ARM based server with Linux OS, in which we enabled the characterization of 72 DRAM chips under relaxed refresh period and various temperatures controlled by a unique thermal testbed. We evaluate the DRAM reliability running single- and multi-threaded HPC workloads on such a commodity server with a fully-fledged Linux OS and a typical multilevel memory hierarchy. In fact, our results show that the manifested Word-Error-Rate under relaxed refresh period varies among the workloads and can be different from the one estimated by the few known fixed data-patterns that were conventionally used in all existing studies. We also discover that the error rates incurred by the execution of the HPC workloads may vary within a program run. Finally, our study shows that the refresh period can be relaxed by 35× leading to 11.2 % power savings on average, while avoiding any system disruption, since the available error-correcting-codes were able to correct all incurred errors up to 60 °C.

ACM Reference Format:

Lev Mukhanov, Konstantinos Tovletoglou, Dimitrios S. Nikolopoulos, and Georgios Karakonstantis. 2018. Characterization of HPC workloads on an ARMv8 based server under relaxed DRAM refresh and thermal stress. In *Proceedings of SAMOS conference (SAMOS'2018)*. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SAMOS'2018, July 2018, SAMOS, Greece

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.

<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

The rapid growth of Internet-of-Things is estimated to generate 24.3 exabytes of data [4] by 2020, creating immense needs for more data storage and aggressive scaling of DRAMs. Such needs have already turned DRAM based subsystems into one of the main power consumers, especially in servers, with estimations indicating that soon they will be accountable for almost half of the overall consumed power [5].

This reality has attracted the attention of many studies that tried to reduce the DRAM power, estimated to incur 40 % power overheads in future 64Gb densities [12]. The majority of existing schemes either try to identify weak cells through offline characterization and adopt a low refresh period only for them [2, 10, 12] or exploit various error mitigation schemes [8, 18] to correct any error induced under a relaxed refresh period. However, the majority of existing schemes [8, 12, 18] are being evaluated based on worst-case error rates estimated by experimental studies performed on FPGAs [7, 11] using few known worst-case data-patterns (DPs). By doing so such studies may over- or under- estimate the manifested errors since they neglect many workload-dependent system-level factors. For instance, the actual data stored on memory during the execution of single- or multi- threaded applications on servers may dynamically differ from the worst-case DPs used conventionally. Furthermore, the number and the frequency of accesses may be such that can inherently refresh the stored charge [1, 23] and thus avoid any error even under relaxed refresh period. Few recent works have tried to exploit inherent error-resilient features of applications [7, 13], but they have done so on simulators and were not able to capture the system-level effects of servers, their multilevel memory hierarchies and the operating system. Related studies have also shown that the cell retention time changes dynamically [20, 21]. Therefore, the offline DRAM characterization performed by current schemes that assume that weak cells will remain fixed after deployment irrespectively of the executed workload is insufficient [7, 13, 19]. To summarize, there is a need to investigate the DRAM behavior under relaxed refresh, while considering the impact of all workload-dependent factors within a real server and evaluate the efficacy of the available error mitigation schemes along with the potential power savings on server-grade DRAMs.

This paper addresses the aforementioned challenges and makes the following contributions:

- We develop a novel experimental framework for characterizing DRAMs under relaxed refresh period within a state-of-the-art

64-bit ARM based server with a Linux OS. In order to experiment under different DRAM temperatures, we also implemented a thermal testbed that allows to fine tune the temperature of each DIMM on the server.

- We perform characterization of the power-reliability trade-off of 72 server grade DRAM chips under scaled refresh period using conventional data-pattern micro-benchmarks (DPBench) as well as single- and multi-threaded HPC workloads. Our study reveals that the DRAM error behavior varies among the workloads and within execution of these workloads.
- We demonstrate that the total memory power could be reduced by 11.5 % on average in such server with a complete software stack by relaxing the refresh period by 35%. This was possible without compromising the system availability since Single-Error-Correction Double-Error-Detection Error Correction Code (SECCED ECC) was adequate for correcting all manifested errors up to 60 °C, while avoiding any system crashes.

The rest of the paper is organized as follows. Section 2 describes the background and the open challenges, while Section 3 presents our experimental framework. Section 4 analyses the results of our experimental campaign. Finally, conclusions are drawn in Section 5.

2 DRAM BACKGROUND AND CHALLENGES

A main memory sub-system based on DRAMs is organized hierarchically into channels supporting a number of DRAM modules. Each Dual In-line Memory Module (DIMM) usually has two ranks, each of which consists of DRAM chips. Within each chip, DRAM cells are organized into banks, which are two-dimensional arrays, addressed based on rows and columns as shown in Figure 1. The main drawback of the DRAM technology is the limited *retention time* [11] of the cell's charge. To avoid any error induced by the charge leakage over time, DRAM employs an *Auto-Refresh* mechanism that periodically recharges all cells in the array based on the worst retention time across them. Conventionally, all DDR technologies adopt today a *refresh period*, T_{REFP} , of 64 ms for refreshing periodically each cell of the DIMM even if in reality many cells may have a much higher retention time than T_{REFP} and the conditions in the field after deployment may not be as bad as the ones assumed [7].

2.1 Existing Studies

Several experimental studies have revealed the large spatial distribution of cells based on their retention time and tried to exploit it for relaxing T_{REFP} for most of the cells [10, 12, 18]. Typically, such approaches rely on the characterization of the retention time of cells on custom FPGA based setups using i) a set of worst-case DP-Bench and ii) elevated temperatures, which are the main factors that excite worst-case circuit-level effects on DRAMs [8, 10, 11, 19].

DPBench. Typically, a set of worst-case static (all 1's, all 0's, checkerboard) and dynamic (random, uniformly distributed data) DPs are used during characterization since the retention time of each cell depends on the value of the stored data within each cell but also in the neighbouring cells. In fact, recent studies have shown that various circuit-level crosstalk effects are being excited by the stored data and may cause the retention time of each cell to vary [11]. Such findings deemed ineffective any existing approach

that relied on offline characterization since the number and position of the identified weak cells could not be guaranteed after the DRAM deployment.

Temperature. Indeed many studies have shown that the retention time of DRAM cells decreases exponentially as the DRAM temperature T rises, described by $Ae^{-0055 \times T} + C$ [8, 11, 19]. In all existing studies the DRAM temperature is controlled by placing the whole FPGA system in a thermal chamber [8, 10, 11, 19], or more lately by using heating elements on DIMMs [7].

To address such effects some recent studies suggested the use of error correction codes [10, 18], strengthening the conventional SECCED ECC [6]. Alternatively, other works have tried to mask any error by exploiting the inherent error-resilient features of some applications or the implicit refresh incurred by every memory access but they did so mainly within simulators [1, 13], rather than on a real system, basing their fault injection schemes on the retention time discovered by the DPBench.

2.2 Challenges and Objectives

Prior existing studies have revealed various circuit level data- and temperature- dependent effects and suggested error mitigation methods. However, they still left a number of questions unanswered. First, existing studies were performed on custom FPGA setups or simulators, which may help simplify characterization and evaluation, but do not study the impact of system level effects, which may be excited within a server and affect the DRAM reliability, such as reuse time in [1].

Several parameters of the deep memory hierarchies on servers, like the organization, the size of the caches and the supported memory bandwidth, can directly affect the number and frequency of accesses to DRAM and thus its reliability. The thorough study of the DRAM error behavior requires the execution of relevant single- and multi- threaded workloads on servers. Furthermore, real workloads are expected to dynamically change the stored DPs and load different components of a system and thus may cause various system level effects that may not excite the worst-case scenarios targeted by conventional DPs.

As temperature has a significant effect on the DRAM reliability [8, 11, 19], it is also essential to develop a special thermal testbed and investigate the DRAM reliability running real workloads under high temperatures, which may be obtained in the worst case scenarios.

Finally, there is a need to investigate if the available ECC on server grade DRAMs or more strong ECCs [6] are required to ensure

Figure 1: X-Gene2 Server and DRAM Hierarchy

non-disruptive operation of such a complex system with a fully-fledged OS, and measure the achieved power savings under relaxed T_{REFP} .

3 EXPERIMENTAL FRAMEWORK

To address the aforementioned challenges, we develop an experimental framework on a server as described below.

3.1 Server Details

The basis of our experimental framework is a state-of-the-art commodity 64-bit ARMv8 based server, the X-Gene2 Server-on-a-Chip, which is the latest generation of the X-Gene family of chips used in the popular HP Moonshot servers [22]. 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 four PMDs 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 Controller Units (MCUs). Each MCU has one channel of DDR3 memory and support up to two DIMMs with two ranks each. In our campaign, we are experimenting with 4 Micron DDR3 8GB DIMMs at 1866 MHz [15], one DIMM per MCU. In total, we are characterizing 72 chips of 4Gb x8 DDR3 [14], since each DIMM includes 16 and 2 DRAM chips for data storage and ECC, respectively.

The X-Gene2 provides access to a separate Scalable Light-weight 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 memory errors corrected or detected by SECDED ECC, providing information about the DIMM, bank, rank, row and column that the error occurred. The available ECC can detect and correct single-bit errors in a 64-bit word, which we refer to as *correctable errors* (CEs) and detect two-bit errors that cannot be corrected, which we refer to as *uncorrectable errors* (UEs). Finally, SLIMpro allows to configure the parameters of the MCUs, such as timings and T_{REFP} , specifically from the nominal 64 ms to 2.283 s that is the maximum allowed T_{REFP} in the X-Gene2 server. 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.

3.2 DRAM reliability analysis

We use the aforementioned error reporting mechanisms to record the CE and UE manifested within each 64-bit word under relaxed T_{REFP} . In addition, to account for any potential errors of more than two-bits in a 64-bit word that cannot be detected by ECC, we compare the output of each execution with a golden reference output obtained when DRAM is operating under the nominal T_{REFP} . In this way, essentially we were able to measure any Silent Data Corruption (SDC) that could go undetected by SECDED ECC.

At the end of the experimental campaign, we analyse the results and quantify DRAM reliability using a set of metrics.

We calculate the 64-bit error rate, WER , for the amount of memory used by an application as:

$$WER(X) = \frac{\#ofUniqError(X)}{SizeOfMemoryUsedInWords} \quad (1)$$

WER shows the probability of a bit being erroneous independently of the memory size allocated by the application.

To compare any differences between the number of error-prone locations discovered by the DPBenchs and the HPC workloads, we calculate the so-called coverage of unique erroneous locations detected when running a specific workload over the total number of error-prone locations, discovered by all benchmarks, as:

$$Cov(X) = \frac{\#ofUniqError(X)}{\sum_{i=bench}^{\forall} \#ofUniqError(i)} \quad (2)$$

, where $\#ofUniqError(j)$ is the number of unique erroneous locations in terms of 64-bit words discovered when running the specific benchmark j . Furthermore, we calculate the rate of change of the Cov in time, as $\Delta Cov(X, t) = Cov(X, t) - Cov(X, t - T_{step})$, where $T_{step} = 10 \text{ minutes}$. This allow us to investigate how fast each benchmark discovers error-prone locations and when Cov converges to a stable value, which can be used as an indicator that we reached an acceptable number of experiments.

3.3 DRAM Thermal Testbed on a Server

To perform the experiments under controlled temperature, we implement a temperature-controlled testbed for DRAMs on a server. Our approach is based on heating elements, similar to [7]. 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 which contains eight solid state relays controlling the resistive elements of each DIMM and rank independently. During our experiments, the maximum deviation from the set temperature is less than $\pm 1^\circ\text{C}$.

4 EXPERIMENTAL EVALUATION

In this section, we present the results of our characterization campaign of 72 DRAM chips using the described framework under the maximum allowed T_{REFP} , i.e. 2.283 s, in the X-Gene2 server at 50°C and 60°C .

4.1 Considered Benchmarks

We choose to run the conventional DPBenchs based on: *all 0s*, *all 1s*, *checkerboard* and *random* DPs as in most existing studies [9, 11]. We select also 4 HPC applications from the Rodinia Benchmark Suite, which are typically used for benchmarking parallel systems [3]. In particular, we use *backprop*, *nw*, *srad* and *kmeans* to cover a range of domains, i.e. Machine Learning, Bioinformatics, Image Processing and Data Mining. We run HPC benchmarks with 1 and 8 threads to evaluate how parallel execution affects the DRAM error behavior.

In our study, we are trying to understand whether DRAM operating under T_{REFP} could be characterized with DPBenchs. To

Figure 2: Comparison of *Cov* across DPBenchs and HPC benchmarks.

investigate this, we need to identify all cells that have a small retention time, and thus are more prone to failures. This is achieved by executing DPBenchs ensuring that we cover all memory available for the user space, i.e. 28 GB on our experimental setup, while the remaining 4 GB is used by Linux for the kernel, drivers and other system services.

For HPC workloads, we limit the datasize of the benchmarks to 8 GB, while each execution has a random allocation in memory as it goes through the normal allocation policies of Linux. We intentionally do not use all available memory at the user space to consider in our study a possible effect of these policies on the DRAM reliability and explore ability of DPBenchs to cover error-prone locations discovered when running real workloads.

4.2 Characterization with DPBenchs

We start with running each one of the considered DPBenchs allocating 8 and 28 GB memory for 120 minutes to ensure that each DPBench makes at least 16 rounds, as discussed in [11]. In all our experiments with DPBenchs under 50 °C and 60 °C, we observe only CEs, which were corrected by the available SECDED ECC, and no UEs or SDCs.

4.2.1 Coverage. In order to identify if the different DPBenchs excite different error locations for DRAM operating at $2.283 \text{ s } T_{REFP}$, we calculate *Cov* of error locations discovered by each DPBench, as seen in Figure 2a and 2b. The highest *Cov* is observed for the *random* DPBench which is above 70 % for both temperatures, while the static DPBenchs discover less than 25 % of all reported error-prone locations. These results are consistent with the observations made by Liu [11], where authors also reported that the highest *Cov* is observed in case of the *random* DPBench. In Figure 2a and 2b, the circle's size and the number indicates the $\Delta Cov(10 \text{ minutes})$ for the last 10 minutes. We see that the $\Delta Cov(10 \text{ minutes})$ does not exceed 2 % for each DPBench at the end of the execution which implies that 120 minutes are sufficient for identifying the majority of error-prone locations. Note that for the calculation of *Cov* above, we only consider the errors incurred by the DPBenchs.

4.2.2 WER. We calculate *WER* for the DPBenchs as can be seen on Figure 3. Similarly to our experiments with *Cov*, we observe that the *random* DPBench does have the highest *WER* compared to

checkerboard, *all 0s* and *all 1s*, specifically 6.7×10^{-7} and 1.2×10^{-5} at 50 °C and 60 °C, respectively. Comparing *WER* of DPBenchs in case of 8 and 28 GB memory allocations, we get similar *WER*s for both temperatures. This shows that *WER* does not change with the size of the allocated memory, which is also an indication of the small variation of manifested errors and weak cells across the considered DIMMs.

4.3 Characterization with HPC benchmarks

Next, we execute single- and multi-threaded versions of the memory intensive HPC benchmarks under relaxed T_{REFP} for 2 hours as in case of DPBenchs. Similarly to the DPBench experiments, we observe only CEs, which were corrected by the available SECDED ECC, and no UEs or SDCs

4.3.1 Coverage. Figure 2c shows *Cov* for DPBenchs and the HPC benchmarks for DRAM operating under relaxed at T_{REFP} 50 °C and 60 °C. In this figure, we note the number of used threads after the name of each HPC benchmark. We see that *random* exhibits the highest coverage, 48 % and 51 % (at 50 °C and 60 °C), and essentially excites more error-prone locations than most of the HPC benchmarks. However, it is lower than *Cov* for *random* discovered in the previous experiment when we consider only error-prone locations detected with the DPBenchs (see Figure 2). We also observe that *kmeans(8)* has the highest *Cov* among HPC benchmarks at 50 °C, which is even slightly higher than *Cov* achieved by the *random* DPBench allocating 8GB of memory. These results imply that *real workloads induce errors in more locations in comparison to the static DPBenchs and, in some cases, even more than the random benchmark*. Finally, as no applications demonstrate high *Cov*, we conclude that each benchmark can trigger errors in memory locations which are not detected when running other benchmarks.

4.3.2 WER. The obtained *WER*s for HPC workloads are depicted in Figure 3. We observe that the HPC benchmarks incur a higher *WER* than the static DPBenchs, but less than the *WER* manifested by *random* and *random(28GB)* at 60 °C. Nonetheless, *WER* incurred by *kmeans(8)* 50 °C is higher than *WER* obtained for *random* benchmarks allocating 8GB and 28 GB. As follows, *real benchmarks may trigger errors in memory locations which are not covered by DPBenchs which implies that DRAM cannot be fully*

Figure 3: WER of the DPBenchs and HPC benchmarks.

characterized with DPBenchs. We also see that WER vary across benchmarks: for example, WER incurred by *nw* is 2.5x higher than WER obtained for *kmeans*(8). What is more, we found that WER incurred by *kmeans* at 50 °C grows in 2.77x if we run it in parallel(8 threads). Thus we may conclude that the DRAM error behavior is workload-dependent and it may significantly change with the level of concurrency in programs.

4.4 DRAM error behavior variation within workloads

In our study, we also discovered that DRAM error behavior may vary significantly within an application run. For example, Figure 4a shows how the number of CEs induced by *kmeans*(8) changes in time for DRAM operating under $2.283 S T_{REFP}$ at 60 °C. We can identify two different phases in this benchmark: at the first phase the average number of CEs per second is less than 20, while at the second phase this number achieves almost 300 per second. By profiling and analysis of this benchmarks [16, 17], we found that the first phase corresponds to the I/O phase of *kmeans* where the input data for this benchmark is retrieved from a file. While at the second phase, the benchmark processes the input data reading and writing to DRAM intensively which explains the high error rate obtained at this phase. Nonetheless, Figure 4c shows that the average number of CEs per second detected at this phase drops down to 50 for the single-threaded version of this benchmark, while the duration of this phase increases in 7.8x. Note that we also observe variations of the DRAM error behavior within the *nw* benchmark. Figure 4b and Figure 4d depict the number of CEs per second detected for the parallel and 1-thread versions of *nw* respectively. We see two distinct phases with a low and a high error rates. Moreover, similarly to *kmeans*, the error rate is lower for the 1-thread version than for the parallel version of this benchmark.

Based on these grounds, we conclude that the DRAM behavior may vary not only across workloads but also within a workload run.

Figure 5: The average power consumption of the memory, accumulated over all DIMMs.

4.5 Efficacy of ECC

As we discussed, one of the aims of our study is to evaluate if the available ECC is effective in detecting and correcting all errors. Our results demonstrate that SECDED ECC that is anyway available in most server-grade DRAMs, can help maintain the DRAM reliability high even under relaxed T_{REFP} by 35x and help to address the varying weak cells across workloads. Note that we also have not discovered any SDCs when running the benchmarks. In fact, all reported errors were only single-bit errors, which were all corrected by ECC, even when DIMMs were operated under a temperature of up to 60 °C. This indicates that more complicated ECC schemes and their resulted overheads could be avoided while operating DIMMs under such a relaxed T_{REFP} for temperatures up to 60 °C.

4.6 Power gain

Figure 5 depicts the memory power and the reduction of power for each benchmark averaged across the DIMMs and the two temperatures when we relax T_{REFP} . These saving can be attributed to the 35x less refresh operations issued by MCUs, caused by the same change of the T_{REFP} . We observe the greatest reduction of power for *nw*(8) at 27.3 %, while many workloads have reduction close to 10 %. On average, we observe that we can save 11.5 % of the total memory power.

5 CONCLUSIONS

In this paper, we present a comprehensive characterization of 72 DRAM chips under relaxed refresh period and under various temperatures within a commodity server using conventional data-patterns along with a set of HPC workloads. To enable our study, we develop an experimental framework based on a state-of-the-art 64-bit ARM based server integrated with a thermal testbed. We demonstrate that the number of excited error-prone locations vary from workload to workload and may differ from the ones discovered by the conventional data-patterns. Moreover, we show that the DRAM error behavior may also change within a workload run. Finally, we show that the DRAM refresh period can be relaxed by 35x on such a commodity system with all errors being corrected

Figure 4: The number of CEs per second for DRAM operating under 2.283 S_{TrEP} at 60 °C

by the available ECC up to 60 °C DRAM temperatures, resulting in 11.5% power savings.

ACKNOWLEDGMENT

This research has been supported by the European Commission under the Horizon2020 Programme, grant agreement 688540 (UniServer) and 732631 (OPRECOMP).

REFERENCES

- [1] A. Agrawal et al. [n. d.]. RefrInt: Intelligent refresh to minimize power in on-chip multiprocessor cache hierarchies. In *IEEE HPCA 2013*. <https://doi.org/10.1109/HPCA.2013.6522336>
- [2] S. Baek et al. [n. d.]. Refresh Now and Then. *IEEE Trans. on Computers* 2014 ([n. d.]).
- [3] Shuai Che et al. [n. d.]. Rodinia: A Benchmark Suite for Heterogeneous Computing. In *IISWC 2009*. <https://doi.org/10.1109/IISWC.2009.5306797>
- [4] Cisco Systems. 2016. Cisco Global Cloud Index: Forecast and Methodology 2015-2020. (2016). <https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf>
- [5] Howard David et al. [n. d.]. RAPL: Memory Power Estimation and Capping. In *ACM/IEEE ISLPED 2010*. <https://doi.org/10.1145/1840845.1840883>
- [6] X. Jian and R. Kumar. [n. d.]. Adaptive Reliability Chipkill Correct (ARCC). In *HPCA 2013*.
- [7] Matthias Jung et al. 2017. A Platform to Analyze DDR3 DRAM's Power and Retention Time. *IEEE Design & Test* (2017). <https://doi.org/10.1109/MDAT.2017.2705144>
- [8] Samira Khan et al. [n. d.]. The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study. *SIGMETRICS 2014* ([n. d.]). <https://doi.org/10.1145/2637364.2592000>
- [9] M. J. Lee et al. 2010. A Mechanism for Dependence of Refresh Time on Data Pattern in DRAM. *IEEE Electron Device Letters* (2010). <https://doi.org/10.1109/LED.2009.2038243>
- [10] Chung-Hsiang Lin et al. [n. d.]. SECRET: A Selective Error Correction Framework for Refresh Energy Reduction in DRAMs. *ACM TACO 2015* ([n. d.]). <https://doi.org/10.1145/2747876>
- [11] Jamie Liu et al. [n. d.]. An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms. In *ISCA 2013*. <https://doi.org/10.1145/2485922.2485928>
- [12] Jamie Liu et al. [n. d.]. RAIDR: Retention-Aware Intelligent DRAM Refresh. In *ISCA 2012*.
- [13] Song Liu et al. [n. d.]. Flikker: Saving DRAM Refresh-power Through Critical Data Partitioning. *SIGPLAN 2011* ([n. d.]). <https://doi.org/10.1145/1961296.1950391>
- [14] Micron Technology. 2009. DDR3 SDRAM MT41J512M8. (2009).
- [15] Micron Technology. 2015. DDR3 SDRAM UDIMM - MT18JF1G72AZ-1G9 - 8GB. (2015). <https://www.micron.com/parts/modules/ddr3-sdram/mt18jsf1g72az-1g9>
- [16] L. Mukhanov, D. S. Nikolopoulos, and B. R. d. Supinski. 2015. ALEA: Fine-Grain Energy Profiling with Basic Block Sampling. In *2015 International Conference on Parallel Architecture and Compilation (PACT)*. 87–98. <https://doi.org/10.1109/PACT.2015.16>
- [17] Lev Mukhanov, Pavlos Petoumenos, Zheng Wang, Nikos Parasyris, Dimitrios S. Nikolopoulos, Bronis R. De Supinski, and Hugh Leather. 2017. ALEA: A Fine-Grained Energy Profiling Tool. *ACM Trans. Archit. Code Optim.* 14, 1, Article 1 (March 2017), 25 pages. <https://doi.org/10.1145/3050436>
- [18] Prashant J. Nair et al. [n. d.]. ArchShield: Architectural Framework for Assisting DRAM Scaling by Tolerating High Error Rates. In *ISCA 2013*.
- [19] Minesh Patel et al. [n. d.]. The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions. In *ACM ISCA 2017*.
- [20] M. K. Qureshi et al. [n. d.]. AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems. In *IEEE/IFIP DSN 2015*. <https://doi.org/10.1109/DSN.2015.58>
- [21] P. J. Restle et al. [n. d.]. DRAM variable retention time. In *1992 International Technical Digest on Electron Devices Meeting*. <https://doi.org/10.1109/IEDM.1992.307481>
- [22] Gaurav Singh et al. [n. d.]. AppliedMicro X-Gene2. In *Hot Chips, 2014*.
- [23] Konstantinos Tovletoglou et al. [n. d.]. Relaxing DRAM refresh rate through access pattern scheduling: A case study on stencil-based algorithms. In *IEEE IOLTS 2017*. <https://doi.org/10.1109/IOLTS.2017.8046197>