Input Size Effects on the Radiation-Sensitivity of Modern Parallel Processors

Daniel Oliveira, Laercio Pilla, Fernando Fernandes, Caio Lunardi,
Israel Koren, Philippe Navaux, Luigi Carro, Paolo Rech

***Abstract*—In this paper, we inspect the impact of modifying benchmarks' input sizes on parallel processors reliability. A larger input size imposes a higher scheduler strain, potentially increasing the parallel processor's radiation sensitivity. Additionally, input size affects the parallel codes throughput, the number of resources used for computation, and their criticality.**

The impact of input size is experimentally studied by comparing the radiation sensitivity of three modern parallel processors: Intel Xeon Phis, NVIDIA K20, and K40. Our test procedure has shown that parallel threads management significantly affects the device Silent Data Corruption sensitivity. Traditional reliability evaluation methodologies may result is a significant error (up to 200%) in the estimated sensitivity of parallel processors to radiation.

I. Introduction

Parallel processors are nowadays widely used in safety critical applications as the Advanced Driver Assistance System, which increases vehicle safety by analyzing camera or radar signals to detect obstacles and activate brakes to prevent collisions [1]. On aircrafts, parallelism is studied to integrate all the circuitry necessary to implement the collision avoidance system [2]. Efficient parallel processing is capable of compressing images in satellites to reduce the bandwidth necessary to send them to ground [3]. The high computational power of parallel processors combined with their low cost and reduced energy consumption, and flexible development platforms are making them indispensable also in High Performance Computing (HPC) applications.

The reliability evaluation and radiation response of parallel processors is a major concern for safety-critical applications. Additionally, reliability has recently become a design constraint for HPC systems due to their large scale. As a reference, the Mean Time Between Failures (MTBF) of a large-scale system such as Titan (today's second most powerful supercomputer [4]) is in the order of dozens of hours [5]. As we approach exascale, the resilience challenge will become even more critical due to an increase in system-scale [6], [7]. In this scenario, a lack of parallel devices resilience characteristics understanding may lead to lower scientific productivity, lower operational efficiency, and even significant monetary loss [7].

Evaluating the reliability of such devices is challenging due to the extreme parallelism exploited in modern parallel processors. Complex schedulers and dispatchers are required to orchestrate parallel threads, and their reliability should be carefully evaluated. Additionally, the number of active threads depends on the problem size, and so does the imposed scheduler strain, the threads dispatch policy, and resources utilization efficiency. In this paper we demonstrate that, while for traditional CPUs it has been sufficient to test the device when executing representative workloads that ensures a maximum device utilization, for modern parallel processors it is also necessary to vary the benchmark's input size to precisely evaluate their behaviors under radiation.

The contributions of this paper are twofold. First, we demonstrate the importance of selecting a proper input size for testing parallel devices and show the need to increase the input size to stimulate the scheduler and control logic. Second, we present a first experimental comparison of the raw (i.e., with mitigation strategies disabled) reliability of the parallel devices that dominate the HPC market: Intel Xeon Phi and NVIDIA K20 and K40. Xeon-Phi, in fact, acts as an accelerator in Tianhe-2, today's most powerful supercomputer [4] and Trinity, the new Los Alamos National Laboratory's (LANL) cluster. NVIDIA K20 and K40 power two of the top 10 supercomputers, including Titan.

Our experimental evaluation was performed using the accelerated high energy neutron beam available at Los Alamos Neutron Science Center (LANSCE) at LANL. By inducing failures in all the components of the device, including the scheduler, dispatcher, and control logic, our neutron beam experiments provide deeper insights into the resilience characteristics of HPC accelerators that are, otherwise, difficult to obtain.

It is well-known that the throughput and efficiency of parallel devices and processors, in general, depend on the executed code and the input size. While the cross section indicates the sensitivity and criticality of resources involved in computation, it does not correlate them with execution time, throughput or efficiency. Thus, the Mean Workload Between Failures (MWBF) metric is used, in this paper, to evaluate the

This work was partially supported by CAPES foundation of the Ministry of Education, grant A117-2013.

D. A. G. Oliveira, C. Lunardi, P. O. A. Navaux, L. Carro, and P. Rech are with the Instituto de Informática, Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, RS, Brazil (phone : +55 (51) 3308-6806, fax : +55 (51) 3308-7308 email : dagoliveira, clunardi, navaux, carro, prech@inf.ufrgs.br).

L. L. Pilla is with the Departamento de Informática e Estatistica, Federal University of Santa Catarina (UFSC), Florianiópolis, SC, Brazil (phone : +55 (48) 3721 7564 email : laercio.pilla@ufsc.br).

I. Koren is with the Department of Electrical & Computer Engineering, University of Massachusetts, Amherst, MA, USA (phone : +1 413 545 2643 email : koren@ecs.umass.edu).

amount of useful data the device correctly produces.

The remainder of the paper is organized as follows. Section II presents the tested devices, the selected parallel algorithms and input sizes, and describes the evaluation methodology. Section III shows the radiation experimental results, and Section IV compares the MWBF of modern HPC devices. Section V concludes the paper.

II. Evaluation Methodology

In this section we introduce the tested devices, present the selected benchmarks, and describe how they were profiled to select adequate input sizes. We also discuss the importance of varying the input size for a proper parallel devices characterization and describe the adopted experimental procedure.

A. Tested Devices

We experimentally evaluated the reliability of NVIDIA K20 and K40, both including a Kepler GK110b GPU chip [8] and of the Xeon Phi board powered by the coprocessor 3120A [9].

NVIDIA devices are fabricated using 28nm planar bulk technology from TSMC while the Xeon Phi is fabricated using 22nm with the Intel 3-D Trigate transistors. The physical implementations of Intel and NVIDIA devices are extremely different. As circuit level details are proprietary, it is not possible to evaluate the devices low-level resources sensitivity. A direct comparison between NVIDIA and Intel devices is then unfeasible and out of the scope of this paper. We will focus on input size impact on the reliability of NVIDIA and Intel devices, which is related to parallelism management and not to the transistor layout.

The K20 includes 13 Streaming Multiprocessors (SMs), while the K40 has 15 SMs. Both devices can instantiate up to 2048 threads per SM. The Xeon Phi includes 57 physical in-order cores with four hardware threads and 32 512-wide vector registers per core. NVIDIA's and Intel's management of parallel processes are extremely different and may impact both the device efficiency and reliability. NVIDIA, for instance, has a hardware scheduler while Intel relies on a dedicated Operating System (OS) to orchestrate execution. The characterization of the parallel threads management is part of the goal of our test procedure.

Most of our experiments were conducted with the Error Correcting Code (ECC), parity mechanism, and Machine Check Architecture (MCA) disabled. On the K40 and K20 ECC can be completely disabled using the *nvidia-smi* tool [8] while for the Xeon Phi we can disable only MCA error log and double bit error detection (i.e., single bit errors are still being corrected) [9]. This should be taken into account when comparing Xeon Phi and K40 or K20 cross sections presented in Section III. Performing the experiments with reliability mechanisms enabled would impede the gathering of a statistically significant amount of data as ECC reduces the Silent Data Corruption (SDC) rate by about one order of magnitude [10]. Additionally, to find the raw architectures reliability it is necessary to disable mitigation mechanisms. ECC, for instance, could mislead the distinction between SDCs and Crashes. In fact, when the ECC is disabled a double bit error may be masked and will not affect the application's output [11]. The same double bit error will trigger an application Crash when ECC is ON [8]. In contrast, MCA triggers an automatic checkpoint-rollback procedure upon the occurrence of a double bit error. Such a procedure takes at least 6 minutes to complete, impeding radiation experiments. A detailed discussion on ECC and MCA efficiency and efficacy is presented in Section III.

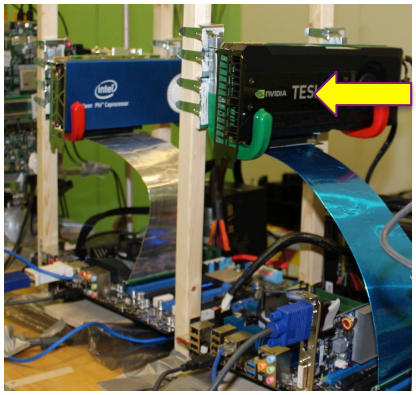

Fig. 1: Part of the experimental setup at LANSCE. Neutrons direction is indicated by the arrow.

B. Tested Benchmarks and Input Size Selection

We select two parallel benchmarks for evaluating the effects of radiation on modern parallel processors: **matrix multiplication (DGEMM)** and **LavaMD**.

DGEMM serves as a cornerstone kernel for several applications and performance evaluation tools. *LavaMD* calculates particle potential and relocation due to mutual forces between particles within a large 3D space. The main computation in this program is the calculation of dot products with floating-point data, where each thread computes the interaction of one particle with all particles in neighboring boxes.

The only constraint that has been traditionally imposed on the input size for processor reliability evaluation, as reported in the literature, has been to excite all available resources [12], [13]. This ensures the evaluation of the worst case sensitivity of a CPU. In fact, increasing the input size would simply mean that the same resource is used more than once, and this affect neither the sensitive area of the device nor the resource criticality [14], [13]. For example, if a cache region is used twice because the data does not fit in the cache, the data the cache holds changes, but the exposed area remains the same. An increase in input size has no impact on the amount of vulnerable resources. Input size changes, in other words, have little to no impact on CPUs' Architecturally Correct Execution (ACE) bits [13].

The situation is different in practice, for parallel processors. Increasing the input size typically increases the number of instantiated threads, too. Depending on the architecture, a higher number of parallel threads can significantly affect the device cross section and execution efficiency.

Threads on Xeon Phi are managed by a dedicated OS, which maps at most four threads to each IMT (Interleaved Multithreading) core. The OS avoids scheduling additional threads on a core as it would lead to context switches and cache conflicts. Additionally, as data is accessed in mostly regular contiguous blocks in the selected benchmarks, cache sensitivity is expected to be unaffected by additional increases in input size. The additional data transfer could affect the efficiency and reliability of the ring that interconnects cores and memory.

NVIDIA uses a different parallel process management for their GPUs. The scheduler and dispatcher, in charge of orchestrating parallel threads execution, are implemented in hardware. A higher number of parallel threads may increase the scheduler strain and, in turn, increase its exposed area. Additionally, data required by active threads that wait to be dispatched is maintained in the large register file available inside the NVIDIA SM. This means that, before being executed, active threads data is exposed and is critical (i.e., an error in active threads data is likely to propagate to the output). By changing the number of active threads, one can also change the time required for a thread to be scheduled again, potentially increasing the exposure time of critical data [15]. Nevertheless, input size increases commonly result in higher throughput [8]. As a result, while increasing the number of threads is likely to increase the GPU cross section, it is also likely to increase the amount of data correctly produced by the GPU and so its MWBF (please refer to Section IV).

To experimentally measure the sensitivity of parallel processors and evaluate the reliability of the parallel threads management, we tailored input dimensions that achieve high resource utilization in the irradiated devices (over 97.5% multiprocessor activity). Stressed resources include register files, cache memories, buses, ALUs, FPUs, control resources, and others. Similarly to traditional CPU tests, in fact, a not fully used resource will show a low cross section not because of a higher reliability but due to a smaller exposed area. Then, to stimulate the peculiar scheduler and dispatcher required in parallel processors, *DGEMM* input dimensions (number of elements in row/column) were varied between 512 and 8192 in powers of two. From 2048 and on, the devices are fully utilized. *LavaMD*'s number of cubes in each dimension of a 3D grid was set to 13, 15, 19, and 23 (each cube contains 100 particles on Xeon Phi and 192 particles on K20 and K40. The number of particles was selected to best fit the hardware).

C. Experimental Procedure

Experiments were performed at the LANSCE facility, Los Alamos, NM, in November 2015. The neutron flux available at LANSCE was about $2.5 \times 10^6 neutrons/(cm^2 \times second)$. Experiments were tuned to guarantee observed output error rates lower than 10^{-3} errors/execution, ensuring a negligible probability to have more than one neutron failure in a single code execution.

The beam was restricted to a spot with a diameter of 2 inches, which was enough to fully irradiate the tested chips without directly affecting nearby board power control circuitry and DRAM chips. This implies that data stored in the main memory is not to be corrupted, allowing an analysis focused on the devices' core reliability.

Figure 1 shows the experimental setup at LANSCE. We irradiate a total of 2 Xeon Phis, K20s, and K40s, placed at different distances from the neutron source. A de-rating factor was applied to consider distance attenuation. After de-rating, the cross section was independent of the position, suggesting a negligible neutron attenuation caused by other boards between the source and the device under test.

A host computer initializes the test by sending pre-selected inputs to the parallel device, collecting results, and comparing them with a pre-computed golden output. When a mismatch is detected, the execution is marked as affected by a *Silent Data Corruption (SDC)*. Software and hardware watchdogs were included in the setup to monitor the application under test, detect application *Crashes*, and perform a power cycle of the host computer in the event of system hang.

III. Experimental Results

In this paper, we report the normalized SDC and Crash cross sections for NVIDIA K20, K40 and Intel Xeon Phi to allow a direct comparison of the considered devices and kernels without revealing business-sensitive data. All values are reported with 95% confidence intervals deriving from Poisson's distribution.

Figs. 2 and 3 show the relative SDC and Crash cross sections for *DGEMM* and *LavaMD*, respectively, obtained increasing the input size. For *LavaMD*, K20 was not tested and K40 with 13 cubes did not provide a statistically significant number of errors. *LavaMD* values for Xeon Phi are multiplied by 10 to allow the inclusion of all the curves in one figure. For these experiments the mitigation mechanisms available on the tested devices were disabled as discussed in Section II-A.

A. Silent Data Corruption

From results reported in Figs. 2 and 3 it is clear that Xeon Phi and NVIDIA devices have a different behavior under radiation, which depends on the executed code. The Xeon Phi SDC cross section seems smaller than the K20 and K40 one for all the codes and configurations but *DGEMM* executed with $2^9 \times 2^9$ double data.

Cross sections are influenced by the different transistor technology and layout (28 nm planar bulk for K40 and 22nm Trigate for Intel). 3-D transistors, in fact, have shown an improved per bit reliability to neutron compared to planar devices [16]. As said in Section II-A, the comparison of the radiation response of the different implementation processes is unfeasible and out of the scope of this paper. Moreover, the different implementation of mitigation solutions intrinsically bias the direct comparison between Intel and NVIDIA (*nvidia-smi* disables completely the ECC, while MCA only disables error logs and double bit error detection). Thus, we limit our discussion to the architectural response of devices.

Even if device resources are saturated the input size has a strong impact on NVIDIA devices cross section but not on Xeon Phi cross section. The only significant increase for the

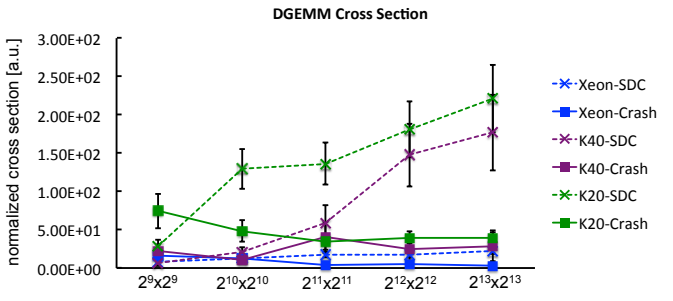

Fig. 2: *DGEMM* normalized cross section.

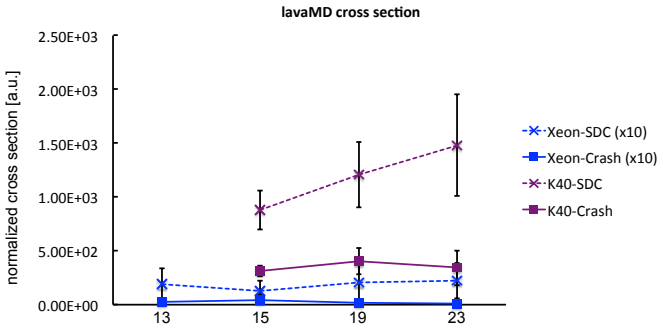

Fig. 3: *LavaMD* normalized cross section (Xeon-Phi cross sections are multiplied by 10 allow the inclusion of all curves in the figure). K20 was not tested and the test on the K40 with 13 cubes did not provide a statistically significant number of errors.

Xeon Phi occurs for *DGEMM* with smaller input sizes (for which the device is not fully utilized). Please notice that all input sizes (but for *DGEMM* executed with $2^9 \times 2^9$) are sufficient to stimulate most of the resources on both devices (see Section II-B). A bigger input size, then, does not increase the amount of resources required for computation and should not affect the cross section (see Section II-B). However, increasing the input size increases the number of parallel threads required for computation. The different behavior between NVIDIA and Intel devices when input size is increased depends mainly on two reasons that derive from the different parallel threads management philosophies.

(1) Increasing the number of parallel threads increases the scheduler strain required to manage and dispatch threads. The scheduler on NVIDIA devices is implemented in hardware and has already been demonstrated to contribute to the device radiation sensitivity [17]. Intel Xeon Phi relies on a dedicated operating system to manage execution [9] which may be less susceptible to radiation-induced failures.

(2) NVIDIA and Intel adopt opposite solutions to manage those threads that are active but waiting to be dispatched. On the K20 an K40, active threads data is kept in registers while other threads are being executed. A larger number of threads increases, then, the time data stays exposed in registers waiting to be used, increasing data criticality and so the cross section. On the contrary Xeon Phi waits for current threads (up to four per core) to finish before launching other ones, so there is no expected cross section increase caused by additional threads.

LavaMD's SDC FIT rate increase with input size is less remarkable than the one seen for *DGEMM* on the K40. For *DGEMM* the cross section is more than doubled from one input size to the next one while for *LavaMD* it is increased of about 30%. This seems to be in contrast with (1) and (2). In fact, *LavaMD* makes heavy usage of local memory (≈14 KB per block of threads), which limits the number of active threads at any given time on the K40. Thus, the increase in number of active threads is limited for *LavaMD*, reducing the impact of (1) and (2).

These observations are possible for parallel devices only if different input sizes are tested. As shown in Figs. 2 and 3, testing only one input size as for a single CPU would result in a significant underestimation of parallel devices cross section. It is worth noting that while the K40 thread management seems to increase its cross section, it may be more efficient. The K40 may then produce more correct data before experiencing a failure. Thread management effects on reliability and throughput give rise to the necessity of considering the MWBF of both devices to draw pragmatic reliability conclusions (Section IV).

B. Crashes

As shown in Figs. 2 and 3, the Crash probabilities are only slightly affected by the input sizes. Even if increasing the input size imposes a higher scheduler strain in the parallel device, the Crash sensitivity remains constant for both devices. As a general result, we can conclude that most of the errors affecting the scheduler do not result in Crash but contribute to SDCs, in accordance with (1) and (2).

An additional insight of our experiments is that the Crash cross sections are found to be almost independent of the executed code for the Xeon Phi while, depending on the code, it changes by more than 1 order of magnitude for the K40. In fact, while Xeon Phi Crash rates are in the order of tens of a.u. for both *DGEMM* and *LavaMD*, K40 Crash rate varies from tens of a.u. for *DGEMM* to hundreds for *LavaMD*. We believe this behavior to be caused by the hardware $versus$ software threads management.

C. ECC and MCA

NVIDIA HPC devices are protected with a Single Error Correction Double Error Detection ECC mechanism, while Xeon Phi includes MCA. Some experiments were performed with MCA and ECC enabled. What we have observed is that both NVIDIA ECC and Intel MCA reduced the SDC rate by about 1 order of magnitude and significantly increased Crashes. The main issue with MCA is that after the Crash the Xeon Phi triggers a checkpoint procedure, whose recovery time may take up to 6 minutes. This is the main reason for not having a good statistic on MCA efficiency. However, the only expected difference in our results for the Xeon Phi when MCA is enabled, would be caused by double bit errors (single

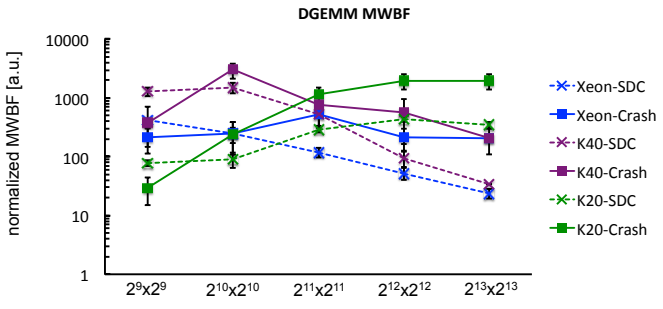

Fig. 4: *DGEMM* normalized MWBF (log scale).

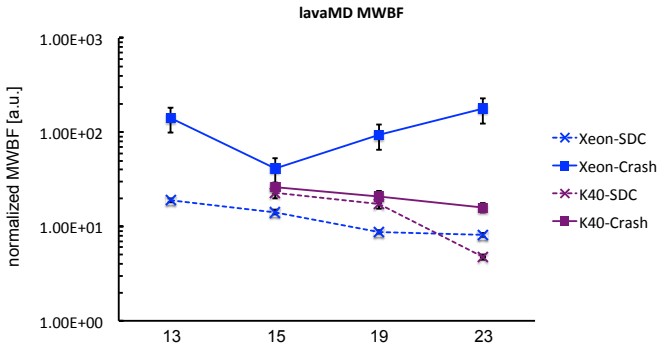

Fig. 5: *LavaMD* normalized MWBF (log scale).

errors are still corrected with MCA off). In contrast, enabling NVIDIA ECC could also lower the effect of input size on the GPU cross section. In fact, the register file would be protected reducing the impact of reason (2) discussed in Section III-A.

IV. Mean Workload Between Failures

In this section, we discuss the Mean Workload Between Failures (MWBF) of Xeon Phi, K20, and K40. The MWBF is the amount of useful work the device produces between failures (SDC or Crash) [18], [17]. The MWBF depends on the cross section (device sensitivity and code criticality) but also on the resources efficiency and throughput. Therefore, we believe the MWBF a more precise reliability metric for parallel processor.

The MWBF is evaluated from the Mean Executions Between Failures (MEBF), which is calculated by dividing the device Mean Time Between Failures by the code execution time. The MEBF is the number of executions correctly completed before experiencing a failure. By multiplying the MEBF by the amount of useful data produced by the device executing the code one obtains the MWBF [17].

Figs. 4 and 5 present the relative MWBF for *DGEMM* and *LavaMD*, for all the tested input sizes. Reported values were normalized to the same value for all the codes, configurations, and devices. From Fig. 4 it is clear that the MWBF of *DGEMM* decreases significantly as the input size increases for the K40 and the Xeon-Phi. The K40 MWBF is higher than Xeon Phi for inputs lower than $2^{12} \times 2^{12}$. However, when the input is equal or greater than $2^{12} \times 2^{12}$, the parallel process management impact is heavier in K40 due to a large number of threads (please refer to the discussion in Section III-A). As a result, the Xeon Phi MWBF becomes comparable with the K40 one. The K20, in contrast, has an increasing MWBF. Regardless the increased cross section caused by the additional scheduler strain (see Fig.. 2), the K20 becomes more reliable as the input size increases. This is because a larger increases the K20 efficiency faster than its cross section.

For *LavaMD* (Fig. 5), the execution time is linear with the workload. Therefore, the MWBF decreases in agreement with the cross section increase. The behavior is similar to the one seen for *DGEMM*, where the MWBF of K40 decreases rapidly with the increase in the number of parallel threads.

The MWBF also highlights the impact of the parallel process management philosophies. To achieve a good efficiency in K20 or K40, regarding performance and reliability, one needs to reach the processors full utilization. Finally, while the number of threads does not significantly impact Xeon Phi's efficiency and reliability, it does for NVIDIA devices. The increase of parallel threads is reliable only if the efficiency improvement it brings is sufficient to compensate the cross section increase.

V. Conclusion

We discuss the importance of using various input sizes to precisely evaluate parallel processors behavior under radiation. Varying the input size, in fact, impacts the scheduler strain, the number of active threads, resource distribution, and resource efficiency. All these effects of input size variation significantly impact the devices' reliability. As experimentally illustrated, the different parallel threads management of the Xeon Phi, K20, and K40 show different radiation behaviors when the number of threads is increased.

References

[1] European New Car Assessment Programme, "Euro NCAP Rating Review, Report from the Ratings Group," June 2012. [Online]. Available: http://www.euroncap.com

[2] J. Becker and O. Sander, "aramis: Project Overview," 2013. [Online]. Available: http://www.across-project.eu/workshop2013/121108_ARAMIS_Introduction_HiPEAC_WS_V3.pdf

[3] European Space Agency, "ESA COROT Mission Documentation," 2014. [Online]. Available: http://www.esa.int/Our_Activities/Space_Science/COROT

[4] J. Dongarra, H. Meuer, and E. Strohmaier, "TOP500 Supercomputer Sites: November 2015," 2015. [Online]. Available: http://www.top500.org

[5] D. Tiwari, S. Gupta, J. Rogers, D. Maxwell, P. Rech, S. Vazhkudai, D. Oliveira, D. Londo, N. Debardeleben, P. Navaux, L. Carro, and A. B. Bland, "Understanding GPU Errors on Large-scale HPC Systems and the Implications for System Design and Operation," in *Proceedings of 21st IEEE Symp. on High Performance Computer Architecture (HPCA)*. ACM, 2015.

[6] R. Lucas, "Top ten exascale research challenges," in *DOE ASCAC Subcommittee Report*, 2014.

[7] M. Snir, R. W. Wisniewski, J. A. Abraham, S. V. Adve, S. Bagchi, P. Balaji, J. Belak, P. Bose, F. Cappello, B. Carlson *et al.*, "Addressing failures in exascale computing," *International Journal of High Performance Computing Applications*, pp. 1–45, 2014.

[8] "NVIDIAs Next Generation CUDA Compute Architecture: Kepler GK110," NVIDIA. [Online]. Available: http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf

[9] "Intel Xeon Phi Coprocessor System Software Developers Guide," Intel. [Online]. Available: https://software.intel.com/sites/default/files/managed/09/07/xeon-phi-coprocessor-system-software-developers-guide.pdf

[10] D. A. G. Oliveira, P. Rech, L. L. Pilla, P. O. A. Navaux, and L. Carro, "Gpgpus ecc efficiency and efficacy," in *International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT 2014)*, 2014.

[11] M. Wilkening, V. Sridharan, S. Li, F. Previlon, S. Gurumurthi, and D. Kaeli, "Calculating architectural vulnerability factors for spatial multi-bit transient faults," in *Microarchitecture (MICRO), 2014 47th Annual IEEE/ACM International Symposium on*, Dec 2014, pp. 293–305.

[12] R. Kost and D. Connors, "Characterizing the Use of Program Vulnerability Factors for Studying Transient Fault Tolerance in Multi-core Architectures," in *Proceedings of the Second workshop on Compiler and Architectural Techniques for Application Reliability and Security*, 2009.

[13] S. S. Mukherjee *et al.*, "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor," in *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*. IEEE Computer Society, 2003.

[14] R. Baumann, "Radiation-induced soft errors in advanced semiconductor technologies," *Device and Materials Reliability, IEEE Transactions on*, vol. 5, no. 3, pp. 305–316, Sept 2005.

[15] G.-H. Asadi *et al.*, "Balancing performance and reliability in the memory hierarchy," in *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, 2005*, ser. ISPASS '05. Washington, DC, USA: IEEE Computer Society, 2005.

[16] J. Noh, V. Correas, S. Lee, J. Jeon, I. Nofal, J. Cerba, H. Belhaddad, D. Alexandrescu, Y. Lee, and S. Kwon, "Study of neutron soft error rate (ser) sensitivity: Investigation of upset mechanisms by comparative simulation of finfet and planar mosfet srams," *Nuclear Science, IEEE Transactions on*, vol. 62, no. 4, pp. 1642–1649, Aug 2015.

[17] P. Rech, L. L. Pilla, P. O. A. Navaux, and L. Carro, "Impact of GPUs Parallelism Management on Safety-Critical and HPC Applications Reliability," in *IEEE International Conference on Dependable Systems and Networks (DSN 2014)*, Atlanta, USA, 2014.

[18] G. Reis, J. Chang, N. Vachharajani, and S. Mukherjee, "Design and evaluation of hybrid fault-detection systems," in *Proceedings of the 2005 International Symposium on Computer Architecture, ISCA'05*. IEEE Press, 2005, pp. 148–159.