High Performance Computing 6th Lecture

OCTOBER 18, 2016

TAKAO KOBAYASHI LAB.

TOSHIYUKI TSUNO

Selected Paper

"Understanding GPU Errors on Large-scale HPC Systems and the Implications for System Design and Operation"

Published in: High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on

Date of Conferrence: February 7-11, 2015

Outline

1: Introduction

2: Background

3: Methodology

4: Understanding and Quantifying GPU Errors on the Titan

5: Experience with Moonlight GPU Cluster

6: Radiation Experiments

7: Conclusion

1: Introduction

Recently, GPUs have evolved from a graphics-specific accelerator to a general-purpose computing device.

Scientists have begun to take advantage of the unprecedented amount of parallelism available in GPUs.

Sientific applications typically employ a checkpoint/restart mechanism, but the efficiency of these mechanisms depends on the “resilience” characteristics of system.

Unfortunately, their resilience characteristics in a large-scale computing system have not been well studied.

2: Background

2.1 GPU Architecture and Resilience Support

SM (Streaming Multiprocessor)

- The thread block scheduler dispatches one or more blocks of threads to an idle SM.
- Each SM has multiple CUDA cores.
- All SMs have access to the shared L2 cache and the device memory.
- The Shared memory and L1 cache regions are a dedicated resource for each SM.
- Each SM has a dedicated register file.

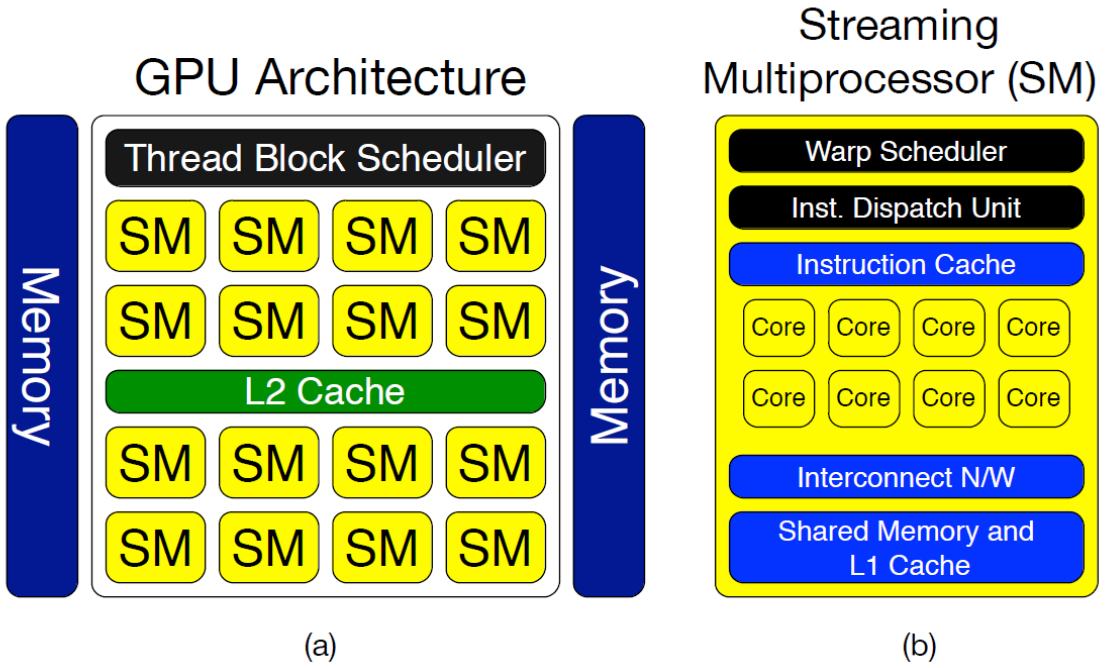

Figure 1: A representative CUDA-based GPU architecture.

2: Background

2.1 GPU Architecture and Resilience Support

CUDA (Compute Unified Device Airchitecture)

- The warp scheduler selects a warp(a group of threads) to be executed next on the CUDA cores.
  (Then the instruction dispatch unit dispatches instructions.)
- Each CUDA core executes only one thread at a time.
- Each CUDA core has access to the shared memory and the L1 cache region.

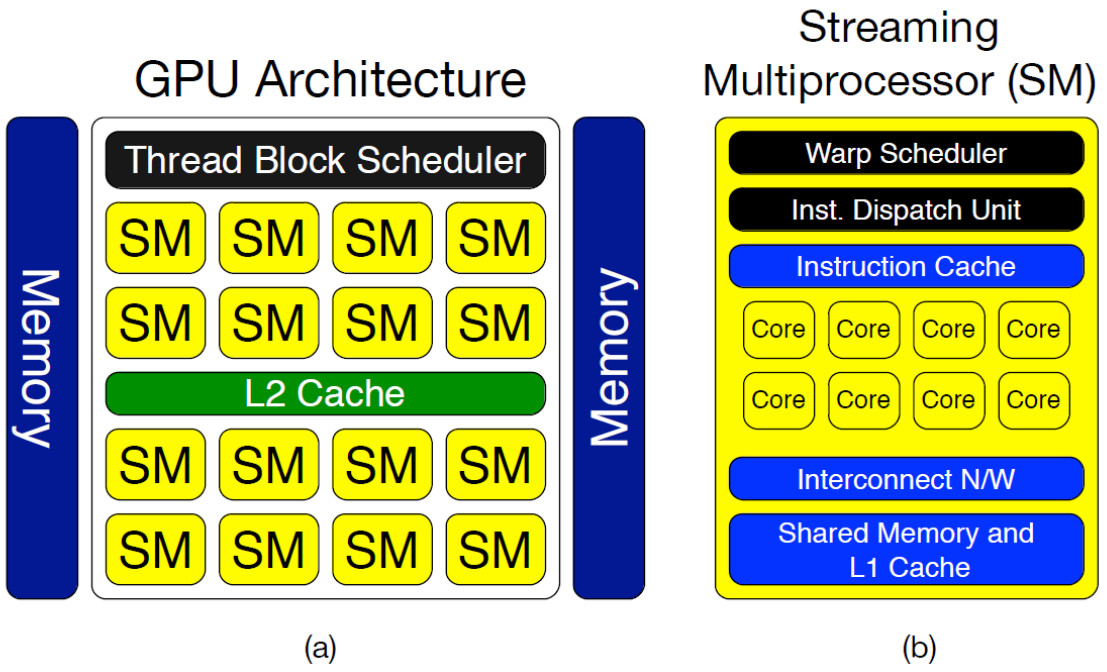

Figure 1: A representative CUDA-based GPU architecture.

SECDED: Single Error Correct and Double Error Detect
ECC: Error Check and Correct

2: Background

2.1 GPU Architecture and Resilience Support

The storage structures

- ex) device memory, L2 cache, inst. cache, register files, shared memory and L1 cache region
- are protected with (SECDED) ECC.

The scheduler structures

- ex) thread block scheduler, warp scheduler, inst. dispatch unit and interconnect network
- are protected with ECC much lower…

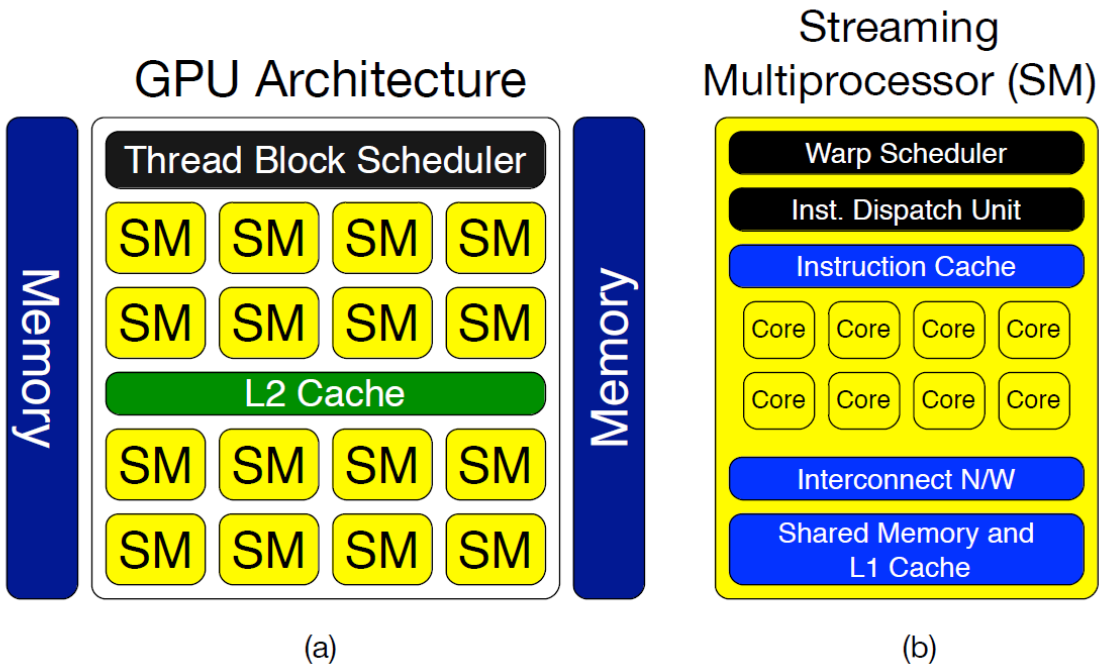

Figure 1: A representative CUDA-based GPU architecture.

2: Background

2.1 GPU Architecture and Resilience Support

The scheduler structures may not be correctly logged by system.

Therefore, in addition to the field data, authors decided to do neutron beam tests, which could find the effect of corruptions in these areas.

2: Background

2.2 GPU Errors and Their Impact

On Table 1, there are the XIDs and impacts of respective GPU errors.

XID: the number of the list on“http://docs.nvidia.com/deploy/xid-errors/index.html”(References[3])
There are the potential causes of the GPU error which has XID.

→The potential cause of the GPU error which doesn’t have XID has been not revealed.
(Single Bit Error and Off the bus)

Table 1: GPU errors and its impact

GPU Error	XID	Impact
Single Bit Error (Corrected by the ECC) Silent data corruption may occur, if no ECC support.	–	No side effect on the program.
Double Bit Error (Detected by the ECC) Silent data corruption may occur, if no ECC support.	48	Program crash.
Off the Bus	–	Program crash
Display Engine error	56	Program crash
Error programming video memory interface	57	Program crash
Unstable video memory interface detected	58	Program crash
Internal micro-controller halt	62	Program crash
ECC page retirement error	63,64	Program crash
Video processor exception	65	Program crash

3: Methodology

3.1 Data Sources and Data Collection Methodology

The authors analyzed the data that were collected from console log.

・The console log has a record for each GPU related event.
(the node location, time-stamp and description of the event)

・Single bit errors are not logged to the console log.

nvdia-smi utility on all the GPU nodes

・Reporting the single and double errors and ECC page retirement related errors

・Providing the information of a particular event

※The authors collected GPU error logs from the Titan Supercomputer and the Moonlight GPGPU cluster.

3: Methodology

3.2 Neutron Beam Test

What used: the neutron flux

→Does it bring about some failures?

No, the probability of more than one neutron generating a failure in a single code execution remains practically negligible.

MTBF: Mean Time Between Failures

4: Understanding and Quantifying GPU…

4.1 Temporal characteristics of GPU failures

From Fig.4, the frequency of GPU related failure events occurred once in 2 days on an average.

This is a significant result(fairly low) in the context of such a large-scale system where more than 2 failures per day are occurred on an average

→It is estimated using vendor-specified MTBF for the GPU card.

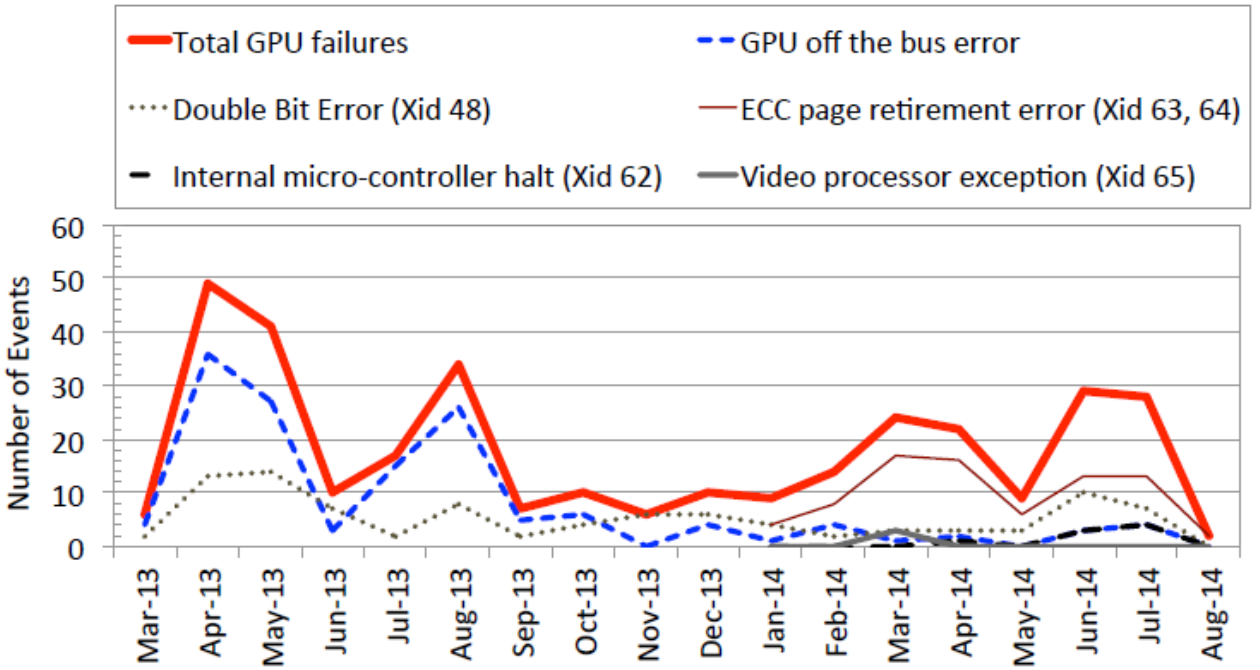

Figure 4: Monthly frequency of different types of GPU failures For the Titan supercomputer

4: Understanding and Quantifying GPU…

4.1 Temporal characteristics of GPU failures

Off the bus failures were dominant only before GPU production run (December 2013).

A system integration issue with GPU cards was identified and resolved by soldering the cards before the system went into production with GPUs.
(We can customize GPU cards before GPU production run.)

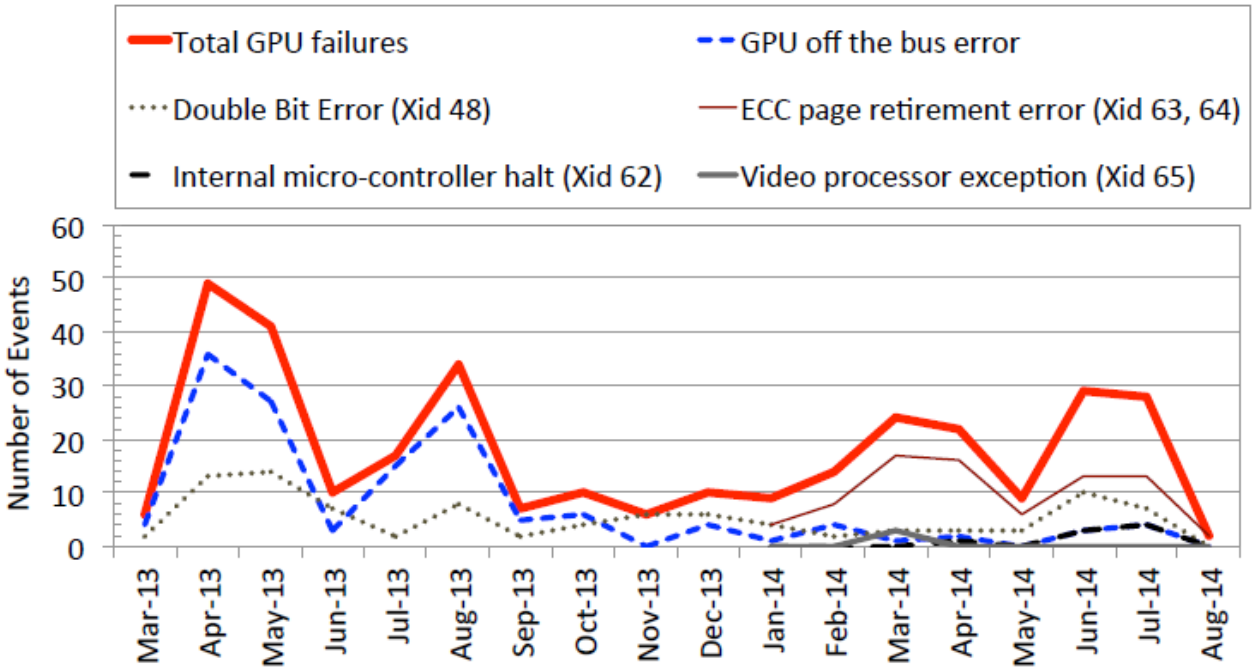

Figure 4: Monthly frequency of different types of GPU failures For the Titan supercomputer

4: Understanding and Quantifying GPU…

4.1 Temporal characteristics of GPU failures

The mean time to application interruption in the production is more than 40 hours, significantly higher than the estimated MTBF of the whole system (11.7 hours).
(using the vendor-specified MTBF for the GPU card)

→Only a small fraction of “bad” GPU cards encounter most of the errors repeatedly, and hence, are enough to bring down the MTBF of the whole system significantly.

So, authors can identify such “bad” cards early and consequently, increase the mean time to application interruption significantly.

4: Understanding and Quantifying GPU…

4.1 Temporal characteristics of GPU failures

From fig.5, a significant fraction of the failures occur much before the observed MTBF. These results indicate that there exists a strong “temporal locality” between GPU failures.

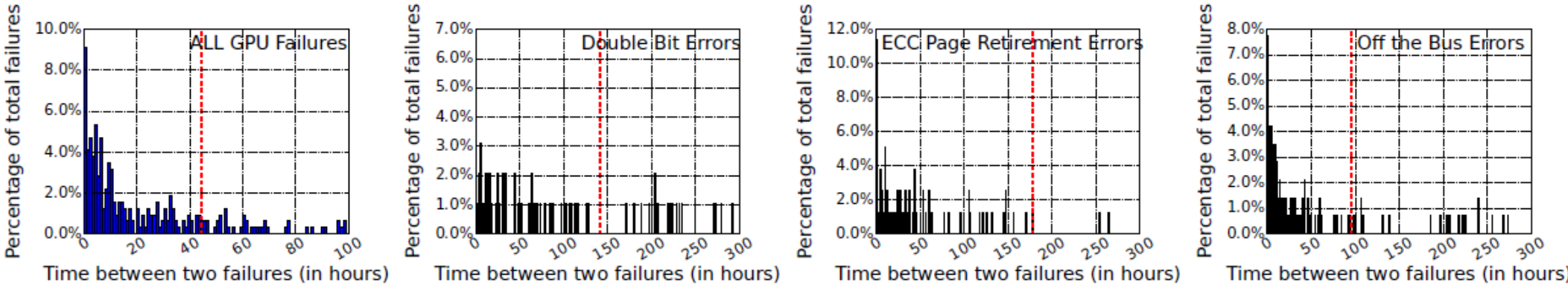

Figure 5: GPU failures exhibit temporal locality: These figures show the failure arrival time distribution. The dashed vertical line indicates the “observed" mean time between failure (MTBF). Multiple failures that occur beyond the x-axis limits are not shown here for clarity, but they contribute toward the MTBF calculation. Note that the combined MTBF is less than the MTBF for each individual types of failure.

4: Understanding and Quantifying GPU…

4.2 Distinct number of cards affected by different failures

How many GPU cards are affected by different GPU failure types.

Authors chose nearly 150 out of more than 18000 cards.

Fig.9 shows that the ECC page retirement and DBE errors affect nearly half the numbers of GPU cards

Moreover, Fig.9 also shows that there are a few cards which experience DBEs multiple times.

(Continuing…)

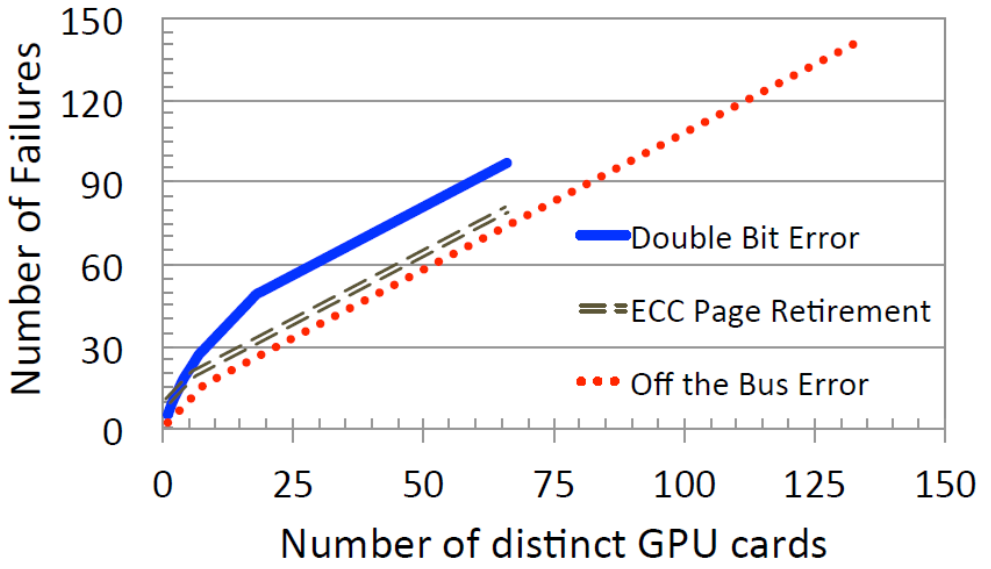

Figure 9: Number of distinct GPU cards affected by different types of GPU failures. A bend in the DBE curve suggests that a few card experience multiple DBEs.

4: Understanding and Quantifying GPU…

4.2 Distinct number of cards affected by different failures

(Continuation of pre-page)

In fact, 6 GPU cards are responsible for 25% of all DBEs.

Regarding ECC page retirement, one particular card is responsible for more than 10% of the ones.

→Certain GPU cards may experience DBEs and ECC page retirement errors multiple times, motivating to identify such cards early.

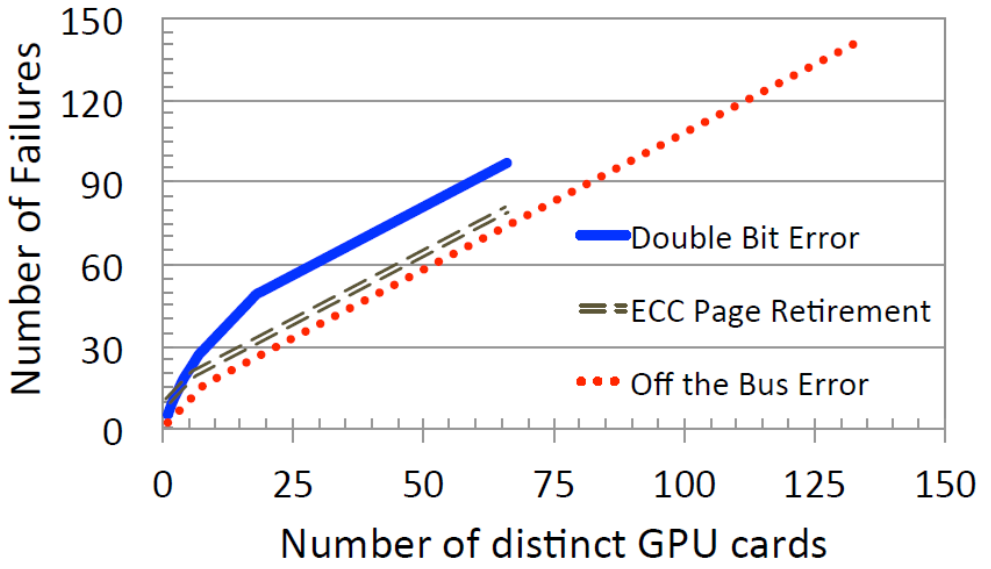

Figure 9: Number of distinct GPU cards affected by different types of GPU failures. A bend in the DBE curve suggests that a few card experience multiple DBEs.

4: Understanding and Quantifying GPU…

4.3 Temperature sensitivity of GPU failures

Authors investigate if GPU failures have sensitivity towards temperature.
The results are as follows on Fig.10.

There are 3 cages, including the same GPU cards.
(Cage1: 27℃ Cage2: 30.5℃ Cage3: 34℃)

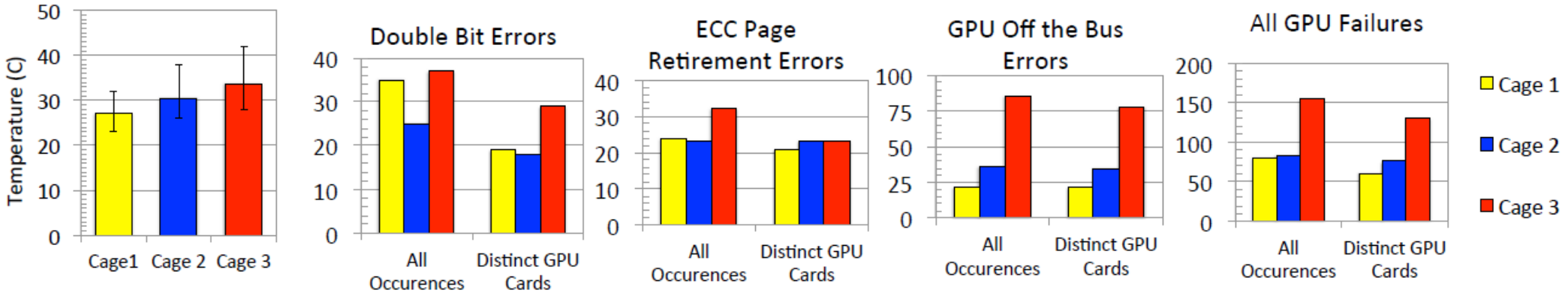

Figure 10: Temperature sensitivity of different GPU failures. Some GPU failure types tend to occur more in relatively hotter cages. "All Occurrences" counts all errors occurring in the same cage. However, in some cases, a single GPU card may see multiple errors. "Distinct GPU Cards" does not count repetitions of an error on the same card.

4: Understanding and Quantifying GPU…

4.3 Temperature sensitivity of GPU failures

The results

- DBEs and Off the Bus may be sensitive to temperature as well.
- ECC page retirement errors are not sensitive to temperature.

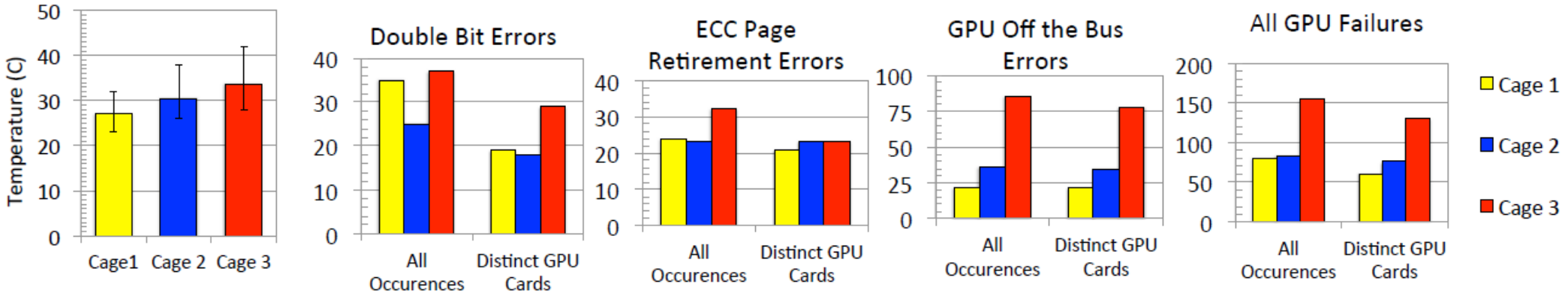

Figure 10: Temperature sensitivity of different GPU failures. Some GPU failure types tend to occur more in relatively hotter cages. "All Occurrences" counts all errors occurring in the same cage. However, in some cases, a single GPU card may see multiple errors. "Distinct GPU Cards" does not count repetitions of an error on the same card.

4: Understanding and Quantifying GPU…

4.3 Temperature sensitivity of GPU failures

However, some cards themselves may be more prone to these errors and variance in the temperature data may further complicate the problem.

Though our field data suggests that some GPU failures may exhibit sensitivity toward temperature, there is a need for more experimental evidence to establish the correlation between GPU errors and temperature with higher confidence…

4: Understanding and Quantifying GPU…

4.4 Analysis of GPU Single Bit Errors (SBE)

※offender = GPU card

From Fig.12, overall the L2 cache region is major contributer in SBE events(showing 98%), and the top 10 offenders also show the same behavior (99% of the SBE events occurred in the L2 cache)

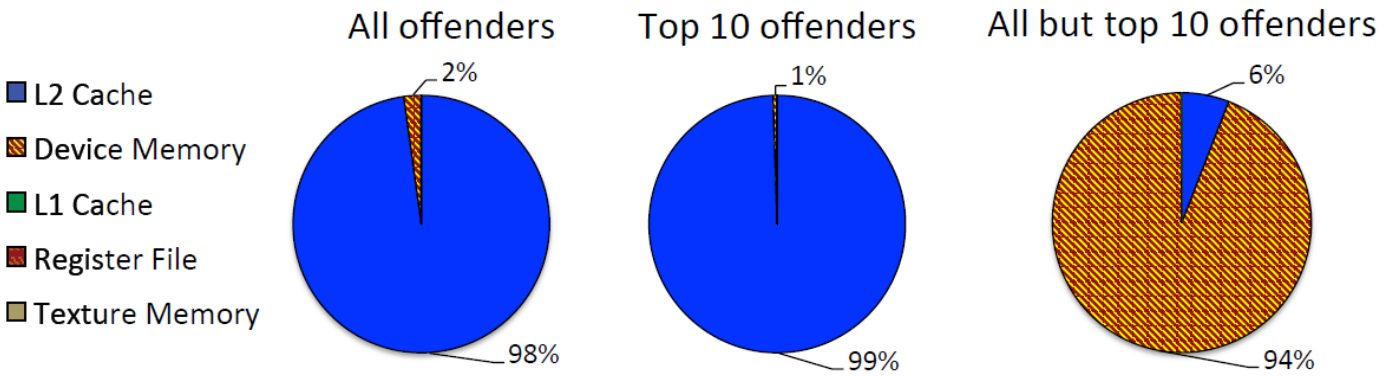

Figure 12: Breakdown of SBEs per storage structure: for GPU cards encountering at least one single bit error.

However, once eliminating the top 10 offenders, the device memory is the structure where most of the SBEs occur (94% of all SBEs).

4: Understanding and Quantifying GPU…

4.4 Analysis of GPU Single Bit Errors (SBE)

GPU cards which experience most of the SBEs are likely to have all the SBEs occur in the device memory instead of the L2 cache.

This finding can be used to identify the top offenders early on.

Our results may also be useful for future architects in terms of which structures need better protection (device memory and L2 cache) and which structures may not need additional costly protection schemes (L1 cache, register file and texture memory).

4: Understanding and Quantifying GPU…

The summary of Section 4

4.1 Temporal characteristics of GPU failures

・ “bad” GPU cards encounter most of the errors repeatedly.

→We can identify such bad cards. And…

・The mean time to application Interruption in the production run is much higher than the estimated MTBF.

・ There exists a strong “temporal locality” between GPU failures.

4.2 Distinct number of cards affected by different failures

・Certain GPU cards may experience DBEs and ECC page retirement multiple times.

→We can identify such bad cards.

4: Understanding and Quantifying GPU…

The summary of Section 4

4.3 Temperature sensitivity of GPU failures

・DBEs and Off the Bus may be sensitive to temperature as well.

・ECC page retirement errors are not sensitive to temperature.

→However, it is solely possible that certain cards are impacted by temperature.

4.4 Analysis of GPU Single Bit Errors (SBE)

・The breakdown of SBEs of both All and Top 10 offenders occur in the L2 Cache.

・The breakdown of SBEs of eliminating the top 10 offenders occur in the device memory.

→Finding that both the L2 Cache and the device memory need better protection.

5: Experience with Moonlight GPU Cluster

5.1 Performance Variation

Authors observed significant performance variation on the GPUs deployed in Moonlight.(M2090 GPUs)

Fig.13 shows that 2 GPUs on the same node exhibit significant performace variation.

This behavior is particularly problematic for HPC workloads which heavily rely on global synchronization. Because of the global synchronization, the overall system performance is determined by the slowest GPU.

→increase in application run time

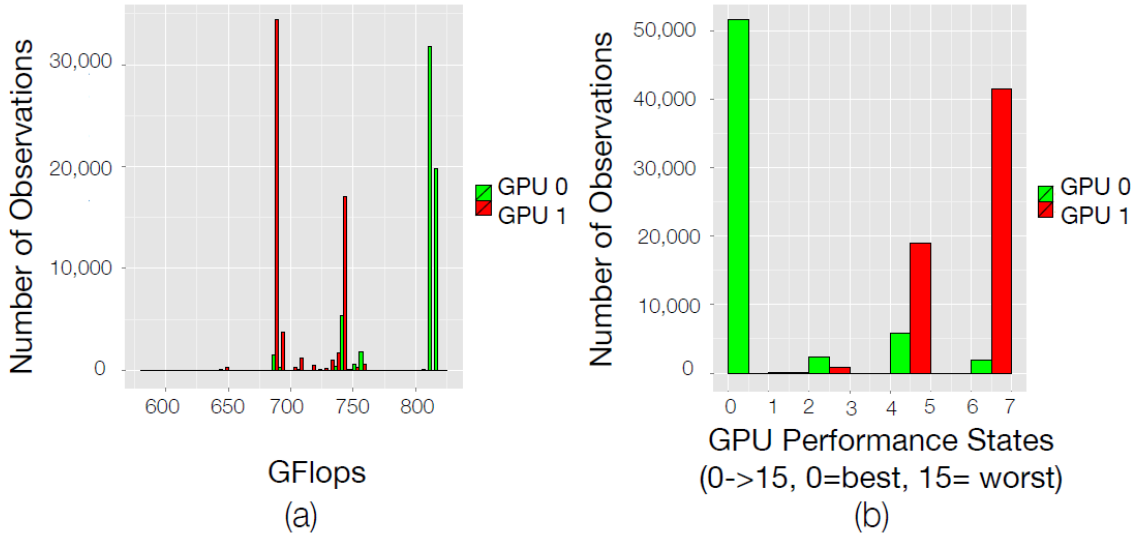

Figure 13: Performance variation on Moonlight's GPU node with two M2090 GPU cards: histogram of observed GFlops (a), histogram of observed performance-states (b).

5: Experience with Moonlight GPU Cluster

5.1 Performance Variation

Fortunately, this problem seems to have resolved in the recently purchased GPUs (K20, kepler architecture). Authors results show less than ±0.02% of performance variation.

5: Experience with Moonlight GPU Cluster

5.2 Inconsistency in Error Logging

Performance variations across GPU cards seem to be fixed on newer generation of GPU cards – making GPUs more amenable to HPC work-loads.

However, inconsistency in error logging (syslog and *nvidia-smi* output) may not have been completely eliminated.

This is critical as system administrators often rely on error counters to monitor the health of the system.

6: Radiation Experiments

6.1 Raw sensitivity of the GPU memory structures

Experimental condition

- GPUs: Kepler architecture based “K20” and Fermi architecture based “C2050” (The K20 structures are significantly larger and newer than the C2050 structures)
- What authors measured: the cross section of the L2 cache and the register file (Cross section = the number of observed errors ÷ (neutrons/cm^2))
- Stored intitial pattern: all “1”s or all “0”s
- What authors expose the device to: a controlled high-energy neutron flux

6: Radiation Experiments

6.1 Raw sensitivity of the GPU memory structures

Fig.14 shows that K20 has reliability than C2050 per bit as well as for the whole L2 cache and register area.

Sgnificant Point

・K20 is larger → better reliability?
(Newer GPUs are ordinarily at lower feature sizes)
→K20 including a better cell design

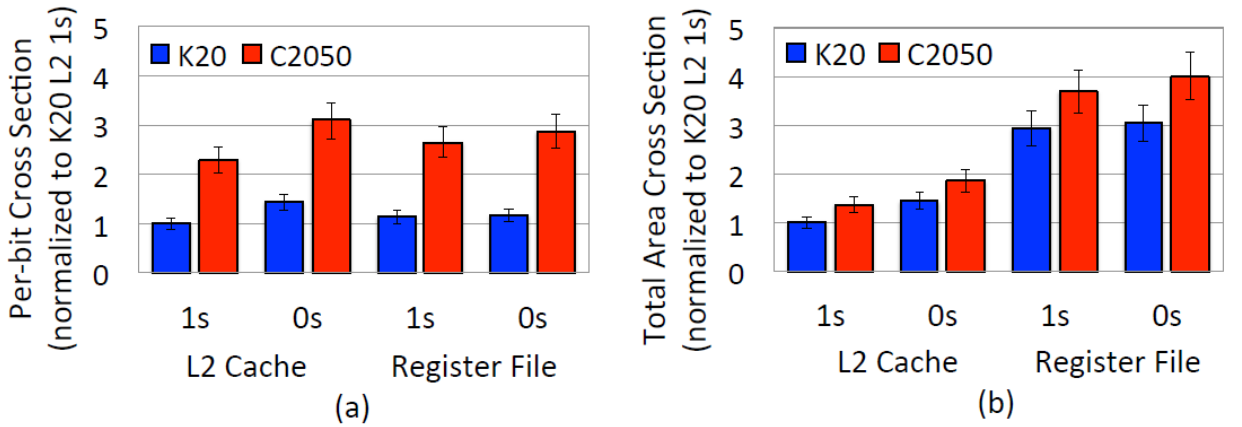

Figure 14: Normalized cross sections for the K20 and C2050 structures: per-bit (a) and total-area (b). Lower is better. The K20 is less prone to bit corruptions than C2050.

・Bits set to “0”s are 40% more prone to corruption than bits in the L2caches for both GPUs.
→This is due to the intrinsic asymmetries to the cache cell design.

6: Radiation Experiments

6.2 Evaluating Radiation Sensitivity of HPC Benchmarks

To evaluate the impact of radiation-sensitivity on real-world HPC applications, authors chose a representative set of benchmarks.

In fact, many real-world workloads often use the algorithms implemented in these benchmarks as their kernel.

On this experiment, authors investigate SDC(Silent Data Corruption) and Crash(a program crash) of the different benchmarks, because SDC and Crash are generated by radiation-induced errors in the GPU memory, logic, or scheduler resources.

6: Radiation Experiments

FIT: Failure In Time

6.2 Evaluating Radiation Sensitivity of HPC Benchmarks

Table 2: Benchmarks details, and FIT at NYC of the different benchmarks (ECC enabled only for the last 2 rows).

	Input	Output	Instr. Exec.	SDC Cross Section	SDC FIT	Crash Cross Section	Crash FIT
lavaMD	1,098,500	878,800	111.3 B	$(2.65 \pm 0.40) \times 10^{-7}$	$(3.44 \pm 0.52) \times 10^{3}$	$(1.17 \pm 0.18) \times 10^{-7}$	$(1.52 \pm 0.23) \times 10^{3}$
FFT	131,072	131,072	30.6 B	$(6.72 \pm 1.01) \times 10^{-8}$	$(8.74 \pm 1.31) \times 10^{2}$	$(4.69 \pm 0.73) \times 10^{-8}$	$(6.09 \pm 0.71) \times 10^{2}$
dgemm	33,554,432	16,777,216	21.8 B	$(6.59 \pm 1.06) \times 10^{-8}$	$(8.57 \pm 1.29) \times 10^{2}$	$(6.27 \pm 0.95) \times 10^{-8}$	$(8.15 \pm 1.22) \times 10^{2}$
MxM	33,554,432	16,777,216	20.4 B	$(5.83 \pm 0.87) \times 10^{-8}$	$(7.57 \pm 1.14) \times 10^{2}$	$(3.34 \pm 0.51) \times 10^{-8}$	$(4.34 \pm 0.65) \times 10^{2}$
Hotspot	2,097,152	1,048,576	266.4 M	$(1.57 \pm 0.24) \times 10^{-8}$	$(2.04 \pm 0.31) \times 10^{2}$	$(8.63 \pm 1.29) \times 10^{-9}$	$(1.12 \pm 0.17) \times 10^{2}$
MTrans	4,194,304	4,194,304	88.1 M	$(2.89 \pm 0.43) \times 10^{-9}$	$(3.75 \pm 0.56) \times 10^{1}$	$(1.55 \pm 0.23) \times 10^{-10}$	$(2.02 \pm 0.30) \times 10^{0}$
MxM ECC	33,554,432	16,777,216	20.4 B	$(5.67 \pm 1.02) \times 10^{-9}$	$(7.38 \pm 1.32) \times 10^{1}$	$(5.09 \pm 0.92) \times 10^{-8}$	$(6.62 \pm 1.19) \times 10^{2}$
Hotspot ECC	2,097,152	1,048,576	266.4 M	$(3.49 \pm 0.52) \times 10^{-9}$	$(4.53 \pm 0.68) \times 10^{1}$	$(1.90 \pm 0.29) \times 10^{-8}$	$(2.47 \pm 0.37) \times 10^{2}$

The difference among *dgemm* and *MxM* is of particular interest as they solve the same problem on the same data. (computing-bounded vs. memory bounded)

dgemm execution, being computing-bounded, requires the GPU cores to be always busy and fully loaded while *MxM* execution the are often waiting for data.
→ the *dgemm* SDC and Crash FIT are higher than the *MxM* ones.

6: Radiation Experiments

6.2 Evaluating Radiation Sensitivity of HPC Benchmarks

Table 2: Benchmarks details, and FIT at NYC of the different benchmarks (ECC enabled only for the last 2 rows).

	Input	Output	Instr. Exec.	SDC Cross Section	SDC FIT	Crash Cross Section	Crash FIT
lavaMD	1,098,500	878,800	111.3 B	$(2.65 \pm 0.40) \times 10^{-7}$	$(3.44 \pm 0.52) \times 10^{3}$	$(1.17 \pm 0.18) \times 10^{-7}$	$(1.52 \pm 0.23) \times 10^{3}$
FFT	131,072	131,072	30.6 B	$(6.72 \pm 1.01) \times 10^{-8}$	$(8.74 \pm 1.31) \times 10^{2}$	$(4.69 \pm 0.73) \times 10^{-8}$	$(6.09 \pm 0.71) \times 10^{2}$
dgemm	33,554,432	16,777,216	21.8 B	$(6.59 \pm 1.06) \times 10^{-8}$	$(8.57 \pm 1.29) \times 10^{2}$	$(6.27 \pm 0.95) \times 10^{-8}$	$(8.15 \pm 1.22) \times 10^{2}$
MxM	33,554,432	16,777,216	20.4 B	$(5.83 \pm 0.87) \times 10^{-8}$	$(7.57 \pm 1.14) \times 10^{2}$	$(3.34 \pm 0.51) \times 10^{-8}$	$(4.34 \pm 0.65) \times 10^{2}$
Hotspot	2,097,152	1,048,576	266.4 M	$(1.57 \pm 0.24) \times 10^{-8}$	$(2.04 \pm 0.31) \times 10^{2}$	$(8.63 \pm 1.29) \times 10^{-9}$	$(1.12 \pm 0.17) \times 10^{2}$
MTrans	4,194,304	4,194,304	88.1 M	$(2.89 \pm 0.43) \times 10^{-9}$	$(3.75 \pm 0.56) \times 10^{1}$	$(1.55 \pm 0.23) \times 10^{-10}$	$(2.02 \pm 0.30) \times 10^{0}$
MxM ECC	33,554,432	16,777,216	20.4 B	$(5.67 \pm 1.02) \times 10^{-9}$	$(7.38 \pm 1.32) \times 10^{1}$	$(5.09 \pm 0.92) \times 10^{-8}$	$(6.62 \pm 1.19) \times 10^{2}$
Hotspot ECC	2,097,152	1,048,576	266.4 M	$(3.49 \pm 0.52) \times 10^{-9}$	$(4.53 \pm 0.68) \times 10^{1}$	$(1.90 \pm 0.29) \times 10^{-8}$	$(2.47 \pm 0.37) \times 10^{2}$

Comparing *MxM* and *Hotspot* to its 2 benchmarks with ECC support.

→ECC reduces SDC rate up to one order of magnitude.

However, a non-negligible amount of SDC may still occur due to errors in unprotected areas (including queues, flip-flops, logics, and schedulers).

6: Radiation Experiments

6.2 Evaluating Radiation Sensitivity of HPC Benchmarks

Table 2: Benchmarks details, and FIT at NYC of the different benchmarks (ECC enabled only for the last 2 rows).

	Input	Output	Instr. Exec.	SDC Cross Section	SDC FIT	Crash Cross Section	Crash FIT
lavaMD	1,098,500	878,800	111.3 B	$(2.65 \pm 0.40) \times 10^{-7}$	$(3.44 \pm 0.52) \times 10^{3}$	$(1.17 \pm 0.18) \times 10^{-7}$	$(1.52 \pm 0.23) \times 10^{3}$
FFT	131,072	131,072	30.6 B	$(6.72 \pm 1.01) \times 10^{-8}$	$(8.74 \pm 1.31) \times 10^{2}$	$(4.69 \pm 0.73) \times 10^{-8}$	$(6.09 \pm 0.71) \times 10^{2}$
dgemm	33,554,432	16,777,216	21.8 B	$(6.59 \pm 1.06) \times 10^{-8}$	$(8.57 \pm 1.29) \times 10^{2}$	$(6.27 \pm 0.95) \times 10^{-8}$	$(8.15 \pm 1.22) \times 10^{2}$
MxM	33,554,432	16,777,216	20.4 B	$(5.83 \pm 0.87) \times 10^{-8}$	$(7.57 \pm 1.14) \times 10^{2}$	$(3.34 \pm 0.51) \times 10^{-8}$	$(4.34 \pm 0.65) \times 10^{2}$
Hotspot	2,097,152	1,048,576	266.4 M	$(1.57 \pm 0.24) \times 10^{-8}$	$(2.04 \pm 0.31) \times 10^{2}$	$(8.63 \pm 1.29) \times 10^{-9}$	$(1.12 \pm 0.17) \times 10^{2}$
MTrans	4,194,304	4,194,304	88.1 M	$(2.89 \pm 0.43) \times 10^{-9}$	$(3.75 \pm 0.56) \times 10^{1}$	$(1.55 \pm 0.23) \times 10^{-10}$	$(2.02 \pm 0.30) \times 10^{0}$
MxM ECC	33,554,432	16,777,216	20.4 B	$(5.67 \pm 1.02) \times 10^{-9}$	$(7.38 \pm 1.32) \times 10^{1}$	$(5.09 \pm 0.92) \times 10^{-8}$	$(6.62 \pm 1.19) \times 10^{2}$
Hotspot ECC	2,097,152	1,048,576	266.4 M	$(3.49 \pm 0.52) \times 10^{-9}$	$(4.53 \pm 0.68) \times 10^{1}$	$(1.90 \pm 0.29) \times 10^{-8}$	$(2.47 \pm 0.37) \times 10^{2}$

Note that Crash increases in case with ECC support.

In spite of ECC support, why does Crash increase?

(Continuing…)

6: Radiation Experiments

6.2 Evaluating Radiation Sensitivity of HPC Benchmarks

(Continuation of pre-page)

ECC has the property “SECDED”(Single Error Correct and Double Error Detect).

DBEs(Double bit errors) are benign, but we can detect DBEs as Crash with ECC.

So, though we consider DBEs as benign errors without ECC, we consider DBEs as Crash with ECC.

6: Radiation Experiments

The summary of Section 6

6.1 Raw sensitivity of the GPU memory structures

・“K20” is newer and larger than “C2050”. Though K20 is larger(Ordinarily, the newer is small.), K20 is better than C2050.
→The reason is that K20 has nice cell design.

・The corruption of bits set to “0”s are particularly big on the cache region.
→Some memory cell designs are intrinsic asymmetric.

6.2 Evaluating Radiation Sensitivity of HPC Benchmarks

・Computing-approaches have more errors than memory-approaches.

・ECC reduces many SDC.

・ECC increases program crashes.
→With ECC, (benign) DBEs are considered as program crashes.

7: Conclusion

The insights about GPU error characteristics can be used to improve operational efficiency of large-scale HPC facilities.

The failure characteristics based on raw sensitivity of the GPU memory structures tested using neutron-beam experiments can be incorporated into the future failure/soft-error modeling, simulation and tool frameworks.

Thank you for your time and attention.