HPC Day 2018

Boston, MA

Northeastern University

NUCAR

Evaluating the Impact of Execution Parameters and Scalarization on GPU Program Vulnerability

Charu Kalra, Fritz Previlon, and David Kaeli

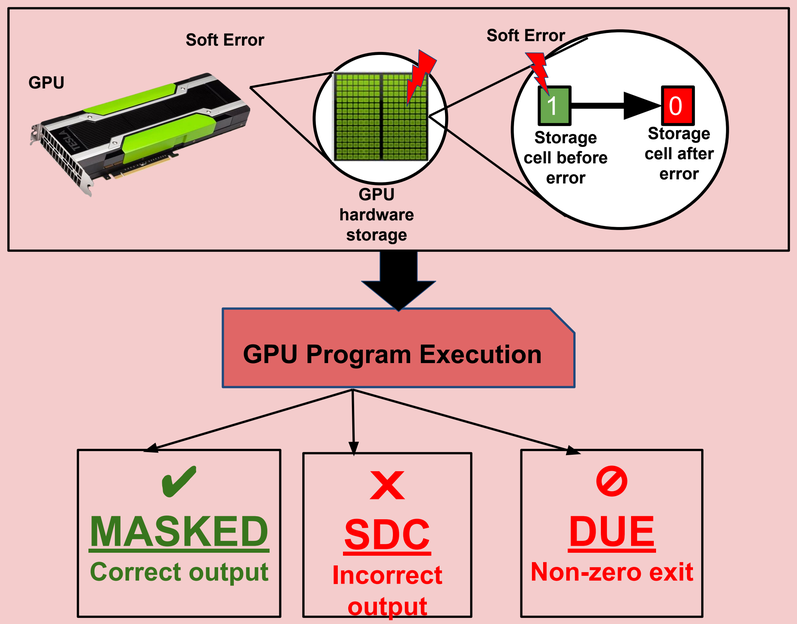
Department of Electrical and Computer Engineering, Northeastern University, Boston, MA

ABSTRACT

- While transient faults continue to be a major concern for the High Performance Computing (HPC) community, we still lack a clear understanding of how these faults propagate in applications.
- This work addresses two particular aspects of the vulnerabilities of HPC applications as run on Graphics Processing Units (GPUs): 1) their dependence on execution parameters (input data and block sizes), and 2) their correlation with specific types of instructions, namely scalar and vector instructions.
- Our results show that the vulnerability of most of the programs studied are insensitive to changes in input values, except in less common cases when input values were highly biased
- We found corruption rate can vary by up to 8% when the block size changes
- Our study also aims to understand the error propagation characteristics when faults occur in scalar, versus vector, instructions
- This analysis will provide insight into potential architectural support required to improve the reliability of scalar instruction execution on GPUs.

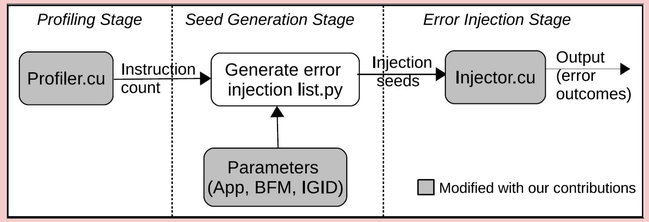
BACKGROUND

Soft Errors in GPUs



FRAMEWORK

- We used SASSIFI fault injection framework for our study on Kepler K20 GPU
- Profiler: Generates instruction count for each instruction type
- Seed Generator: Generates injection seeds based on the given parameters
- Fault Injector: Injects faults in the destination registers based on the seeds



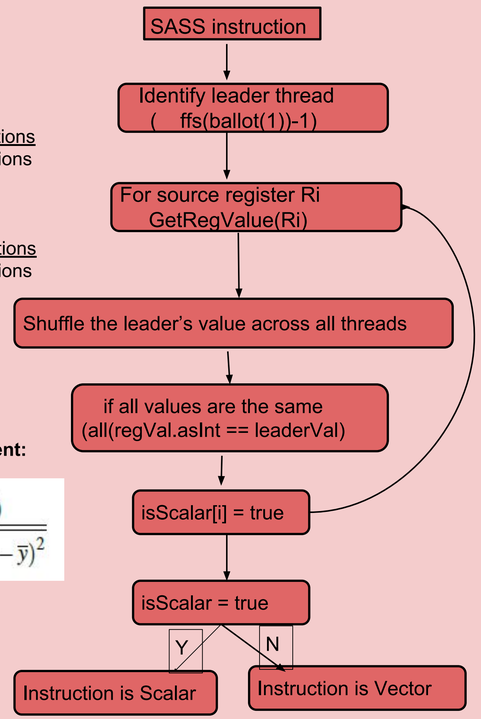
SCALARIZATION

Scalar Intensity =
Num of dynamic scalar instructions / Total Num of dynamic instructions

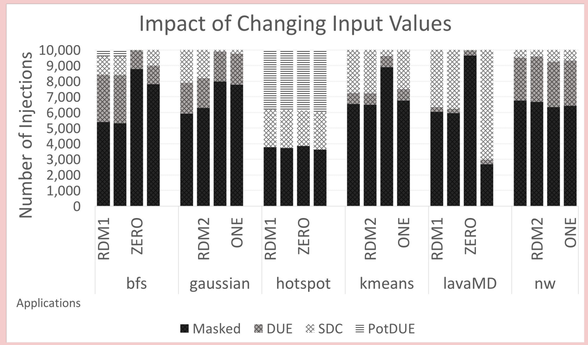
Vector Intensity =
Num of dynamic vector instructions / Total Num of dynamic instructions

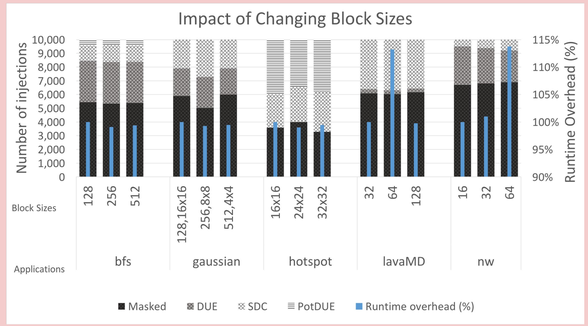
Pearson Correlation Coefficient:

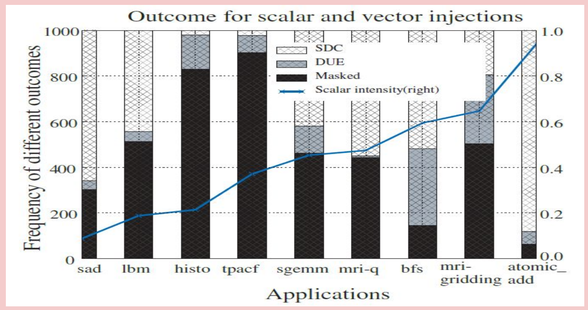
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2 \sum_{i=1}^{n}(y_i-\bar{y})^2}}$$



RESULTS







Vector Opcode	SDC	DUE	Masked
IADD	-0.53	-0.11	0.6
ISETP	-0.56	-0.04	0.6
MOV	-0.58	-0.02	0.6

Scalar Opcode	SDC	DUE	Masked
IADD	-0.55	0.39	0.4
ISETP	-0.6	0.12	0.57
MOV	-0.6	0.06	0.6

CONCLUSION

- The vulnerability of most of the programs studied are insensitive to changes in input values, except in less common cases when input values were highly biased
- The corruption rate can vary by up to 8% when the block size changes
- While some scalar opcodes show positive correlation with the outcomes, others do not.

REFERENCES

1. Fritz G. Previlon, Charu Kalra, David R. Kaeli, Paolo Rech, "**Evaluating the impact of execution parameters on program vulnerability in GPU applications.**" DATE 2018: 809-814
2. Charu Kalra, Fritz Previlon, Xiangyu Li, Norman Rubin, and David Kaeli, "**Analyzing the Vulnerability of Vector-Scalar Execution on Data-Parallel Architectures**" SELSE 2018