Position Paper:
Software Based Techniques for Reducing the Vulnerability of GPU Applications

Si Li, V. Sridharan*, S. Gurumurthi*, and Sudhakar Yalamanchili

School of Electrical and Computer Engineering
Georgia Institute of Technology

+RAS Architecture, AMD Inc.
*AMD Research, AMD Inc.

Scaling Computing Performance

New Packaging

Ultralow Power

Performance

Energy/Power

Reliability

Thermal

Cray Titan: Heterogeneity
Software Reliability Enhancement (SRE)

- Software-based approach to resilience
  - Hardware neutral
  - Not specific to a particular application or algorithm
- Reliability-Aware runtime manager
  - On-line monitoring
  - Apply one or more SRE to meet a reliability/overhead target
  - Recovers or restarts application when an error is detected

*Transparent, customizable, selective, and extensible*

---

SRE: Motivation

- C/C++  OpenCL  CUDA  Python

  **Language Front End**

  **Optimization**

  **Virtual ISA, e.g., HSAIL, PTX**

- **Performance**
  - Application Validation
  - Maintenance and Update
  - Phase Behavior
  - Impact of Deployment Environment

- **Energy/Power**
- **Reliability**

- **One size fits all**
SRE Approach

Key Idea: Code injection and JIT Compilation

Ocelot: Multiplatform Dynamic Compilation

Just-in-time code generation and optimization for data intensive applications

- Environment for i) compiler research, ii) architecture research, and iii) productivity tools
Dynamic Instrumentation: GPU Lynx

Naila Farooqui

Example Instrumentation Code

```
NVCC
PTX
```

- Transparent instrumentation of CUDA applications
- Can treat program level abstractions such as warps and thread blocks


GPU Lynx for AMD

```
__kernel void

test(global int *x)
{
...
}
```

- Prototype with IL
- Moving to HSAIL

Software Reliability Enhancement Framework

- Real time customized information available about GPU usage
- Use this information to drive SRE decisions

Framework: On demand, customizable, transparent, and extensible, software reliability enhancement (SRE)

Past Work: Correctness Checks

- Out-of-Bounds Checks:
  - All kernel memory accesses can be checked against global memory allocations
  - Used to catch out-of-bounds or illegal memory accesses

- Alignment checks
  - Alignment of every memory access checked against instruction type

- Control Flow errors
  - ~3 orders of magnitude faster than using the emulator
Program Vulnerability*

- **Program Vulnerability Factor (PVF)** is a component of Architectural Vulnerability Factor (AVF)
  - It an ISA level measure of how vulnerable the program is to SE
  - It is independent of hardware implementation
  - Code regions with maximum exposure to live state

- Use PVF to direct, online SRE techniques for detection
- Use PVF based analysis to reason about different GPU architectures and applications


**Target of Vulnerability**

Grid of cooperative thread arrays
- Coarse-grain parallelism

PTX Virtual ISA
- RISC Instruction Set
- Defined by NVIDIA - target of CUDA compiler

**Volume of Exposed State**

Cooperative Thread Array
- Fine-grain parallelism
Experiments

- Understand liveness behavior in GPU kernels
- Use this understanding to improve PVF

Insight
- There is a **sweet spot** between low overhead checks and live range size
- Maximize PVF improvement (dynamically) at relatively low cost

Experiment
- Lightweight silent data corruption error detector (Hauberk2011)
- Low-overhead value checksum
- Constant vulnerability & overhead per live-range

Live Range Distribution

![Live Range Distribution Graph](image)
Per Benchmark Instruction Overhead

Preliminary Results

- Reduction in program vulnerability by adding live range checks
Potential Benefits of SRE

- Customizable, extensible, transparent
- Flexible trade-off between runtime overhead and reliability target
- Fault detection and recovery on-demand
- Reduce hardware cost
- End-to-end resilience

Limitations of SRE in BSP

- Compete with kernel for local resources
  - Registers
  - Shared memory
- A single thread cannot terminate kernel
  - If an error occurs in one thread, it cannot abort the kernel
  - Requires hardware support
The Future is (Reliable) Acceleration

Thank You