Sound and Partially-Complete Static Analysis of Data-races in GPU Programs

Dennis Liew[1], Tiago Cogumbreiro[1], Julien Lange[2]
[1]University of Massachusetts Boston, USA. [2]Royal Holloway, University of London, UK

Introduction

GPUs are parallel computation devices with high susceptibility for bugs such as data-races which may produce non-deterministic behaviour.

We proposed *approximation analysis,* a static analysis technique to detect true data-races in GPU kernels, which tests whether accesses are **reachable (Control Independence)**, and when the reported locations are **precise (Data Independence)**.

Our theory was implemented in the tool FaialAA, as the first sound and partially-complete DRF verifier that can detect true data-races.

ALFRED P. SLOAN FOUNDATION

UMass Boston

When are reports from bug finding tools true?

In our evaluation (3 datasets) Faial:

- Reported **2.1× fewer** potential alarms in a dataset of 227 kernels
- Found **12 undocumented** racy kernels, including 8 that are missed by GPUVerify and Faial from a well-studied dataset.
- Certified **5 documented bugs** (OpenMM and Nvidia's Megatron-LM) **and their fixes**, while others only succeed in 2 documented bugs+fixes.

Faial is a static analyzer for CUDA kernels that can check for racy and data-race free kernels.
Try our GitHub Action!

Evaluation

	GPUVerify		Faial		Faial + AA	
Kernel	Racy	DRF	Racy	DRF	Racy	DRF
bucketPos	P-R	DRF	P-R	DRF	T-R	DRF
compRange	P-R	DRF	P-R	DRF	T-R	DRF
reduceVal	P-R	✖	n/a	n/a	T-R	DRF
sortBucket	t/o	t/o	n/a	n/a	P-R	✖
gradInput	✖	DRF	n/a	n/a	T-R	DRF
layerNorm	✖	DRF	n/a	n/a	T-R	DRF

Tools	Kernels				Alarms	
	DRF	P-Racy	T-Racy	Unsupported	True	Potential
GPUVerify	193	17	n/a	16	n/a	50
Faial	207	11	n/a	8	n/a	21
FaialAA	**210**	**4**	**12**	**0**	**22**	**10**

Theoretical Results

Theorem 4.5 (True Positives): identifies a specific class of programs where our analysis only report **true alarms**.

Let $\varnothing \vdash$ CI $[[s]]$, $\varnothing \vdash$ DI $[[s]]$, and *datarace*(δ_1,δ_2).

If $\delta_1 \in$ *p-actions*$([[s]])$ and $\delta_2 \in$ *p-actions*$([[s]])$, then $\delta_1 \in$ *j-actions*(s) and $\delta_2 \in$ *j-actions*(s).

This material is based upon work supported by the National Science Foundation under Grant No. 2204986. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.