

## GPGPU 프로그램의 자료경합 탐지기법을 위한 벤치마크 모음

이건표<sup>0</sup>, 최으뜸\*, 전용기\*

<sup>0</sup>경상대학교 정보과학학과

e-mail: gnrury2@gnu.ac.kr<sup>0</sup>, {slateblue33, jun}@gnu.ac.kr\*

## A Benchmark Suite for Data Race Detection Technique in GPGPU Programs

Keonpyo Lee\*, Eu-Teum Choi\*, Yong-Kee Jun\*

<sup>0</sup>Dept. of Infomatics Gyeongsang National University

### ● 요약 ●

자료경합은 두 개 이상의 스레드가 같은 공유메모리에 적절한 동기화 없이 접근하고, 적어도 한 개의 접근사건이 쓰기일 때 발생할 수 있는 동시성 오류이다. 자료경합은 프로그래머가 의도하지 않은 비결정적인 수행결과를 초래하여, 항공기 소프트웨어와 같은 고신뢰성이 요구되는 프로그램에서 치명적인 오류를 발생시켜 인적 물적 손해로 이어질 수 있다. 자료경합 탐지기법은 이러한 문제를 사전에 탐지하여 수정하는데 사용되어진다. 하지만 GPGPU 프로그램에서의 자료경합은 CPU 병행프로그램에서보다 복잡한 실행구조를 가지고 있어 스레드 및 메모리 계층, 스케줄링, 동기화 기법 등의 많은 변수가 존재한다. 이로 인해 실제 프로그램에 자료경합 탐지기법을 적용하여 검증 시 이러한 변수들을 반영하여 실험하는데 많은 노력이 소요된다. 본 논문은 실제 프로그램에서의 자료경합을 대표하는 4가지 패턴의 합성프로그램으로 이루어지고 실행 시 스레드 및 메모리 계층, 스레드 구조, 메모리 사용량 및 동기화 방안을 지정할 수 있는 벤치마크 모음을 제시한다.

**키워드:** GPGPU, 자료경합(data race), 자료경합 탐지(data race detection), 항공기 소프트웨어(airborne software)

### I. Introduction

병행프로그램에서는 프로그래머가 의도하지 않은 비결정적인 수행 결과를 일으키는 자료경합이 발생할 수 있다. 이로 인한 치명적인 사고를 방지하기 위해 항공기 소프트웨어와 같은 고 신뢰성이 요구되는 분야에서는 자료경합 탐지 기술이 활용된다[1].

하지만 GPGPU 프로그램의 경우 동작 및 메모리 계층으로 인하여 CPU 프로그램보다 자료경합이 발생 및 탐지에 관계되는 조건이 많고, 분석이 어렵다[2]. 따라서 자료경합 탐지 기술의 개발 및 검증과정에 실제 프로그램이나 합성 프로그램을 사용할 경우 변인의 조작과 결과의 분석에 많은 시간이 소요된다.

본 논문에서는 GPGPU 환경에서 발생하는 자료경합의 패턴을 대표하는 합성프로그램으로 이루어지고 옵션에 따라 동기화방안, 스레드 수와 그룹, 메모리 계층 및 사용량, 접근사건 수를 조정할 수 있는 벤치마크 모음을 제시한다.

### II. Preliminaries

GPGPU 프로그램의 자료경합은 GPU의 SIMT (Single Instruction Multiple Thread) 실행 모델에 따라 세분화된다. GPU 프로그램의 스레드는 warp, block, grid 순으로 그룹화 되어 동작한다.

이중 같은 warp에 속한 스레드들은 프로그램 카운터를 공유하며, block은 같은 SM에서 실행되며 shared memory를 공유하는 warp의 집단, grid는 GPU 프로그램을 의미하는 kernel을 수행하는 모든 스레드의 묶음으로 global memory를 공유한다.

GPGPU의 자료경합을 탐지하기 위한 기존의 연구들[2, 3] 실제계에 존재하는 마이크로 벤치마크나 커널에 탐지기법을 적용하여 자료경합이 탐지됨을 검증하고 있다. 하지만 이 경우 여러 가지 유형의 자료경합을 발생시키는 실제 프로그램은 수집해야하며, 자료경합이 발생하는 환경을 변화시키기 위한 프로그램 수정이 어려운 문제가 있다.

### III. Design and Implementation

본 논문은 GPGPU 프로그램의 자료경합 탐지기법의 개발 및 검증과정에서 사용될 수 있는 벤치마크 모음을 제시한다. 이를 위해서는 실제계에서 발생하는 자료경합을 대표할 수 있는 자료경합 패턴을 가진 합성프로그램을 제공하고, 자료경합이 발생하는 변인을 조작할 수 있어야한다.

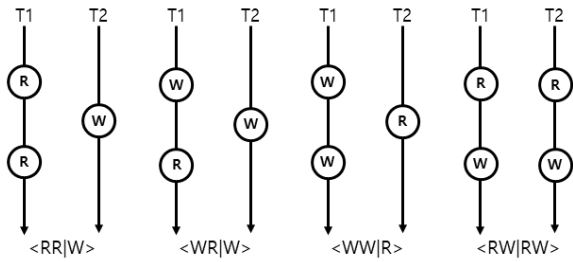


Fig. 1. Patterns of Data Race

합성프로그램은 그림1과 같이 실제 CPU 프로그램의 자료경합을 대표하는 4가지 패턴의 합성프로그램으로 구성된다. 각 패턴은 원자적으로 수행되어야 하는 스레드 1의 두 접근사건 사이에 스레드 2의 접근사건이 발생할 수 있는 경우를 나타낸다.

또한 제시된 패턴이 수행되는 환경을 조작할 수 있도록 한다. 먼저 스레드 계층은 자료경합을 일으키는 스레드의 그룹을 지정하여 같은 intra-warp, intra-block, inter-block에서 자료경합이 발생하도록 한다.

메모리 계층 및 크기는 자료경합이 발생하는 메모리 계층을 shared memory와 global memory중에서 지정하고, 할당하는 메모리 크기를 설정한다.

세 번째로 스레드 구조의 변경은 커널을 실행하는 block 수와 block 이 포함하는 스레드 크기를 지정한다. GPU의 SM 개수에 따라 block과 스레드의 개수는 스레드 인터리빙에 영향을 끼칠 수 있다.

마지막으로 동기화 방식 변경은 접근사건을 barrier, threadfence 또는 atomic function 으로 보호하도록하여 자료경합의 오 탐지를 확인할 수 있도록 한다.

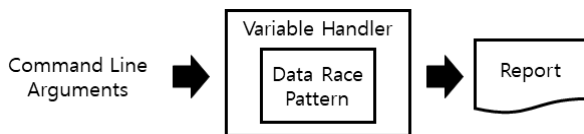


Fig. 2. Design of Benchmark Suite

제시된 벤치마크 모음은 그림 2와 같은 형태로 구현된다. 변인의 조작은 프로그램 실행 시 매개변수 형태로 전달된다. 또한 각 합성프로그램은 소스코드 형태로 제공되어 동적 및 정적분석을 이용하는 탐지기법에서 모두 사용되어질 수 있도록 한다.

#### IV. Conclusions

본 논문은 GPGPU 프로그램의 자료경합 탐지기법의 개발 및 검증과정에서 사용될 수 있는 벤치마크 모음을 제시한다. 이는 실제 프로그램의 자료경합을 대표하는 4가지 자료경합 패턴을 가지며, SIMT에 따른 특성을 반영시킬 수 있다. 이를 통해 자료경합 탐지기법을 다양한 환경에서 검증할 수 있다.

#### ACKNOWLEDGEMENT

이 논문은 교육부 지방대학특성화(CK-I)사업의 재원으로 경상대학교 창의적융합공IT기계융합인력양성사업의 지원을 받아 수행되었음.

#### REFERENCES

- [1] S. Lu, S. Park, E. Seo, and Y. Zhou, "Learning from mistakes: a comprehensive study on real world concurrency bug characteristics," ACM SIGOPS Operating Systems Review, Vol. 42, pp. 329-339, Feb 2008.
- [2] P. Li, X. Hu, D. Chen, J. Brock, H. Luo, E.Z. Zhang, and C. Ding, "LD: Low-overhead GPU race detection without access monitoring," ACM Transactions on Architecture and Code Optimization (TACO), Vol. 14, pp. 9, Jan 2017.
- [3] A. Eizenberg, Y. Peng, T. Pigli, W. Mansky, and J. Devietti, "BARRACUDA: binary-level analysis of runtime RAcEs in CUDA programs," Vol. 52, pp. 126-140, Jun 2017.