

Running GEANT4 Functions on a GPU

Discussion of Results

Stuart Douglas – dougls2
Rob Gorrie – gorrierw
Matthew Pagnan – pagnanmm
Victor Reginato – reginavp

McMaster University

April 8, 2016

Overview

1 Introduction

- Brief Project Overview
- Explanation of Terms
- Scope
- Purpose

2 Features

- All projects can run in CPU or GPU environment
- Able to build GEANT4 with GPU on or off

3 Particle Simulations on GPU

- Entire G4ParticleHPVector Object on GPU
 - Intensive Functions on GPU
 - Performance
 - Accuracy
 - Testing
- Implementation 2
 - Performance

Brief Project Overview

Take an existing particle simulation toolkit - GEANT4 - and have some functions run on a GPU device to improve performance.

Definition: GEANT4

GEANT4 is

What is GEANT4

- Geant4 is a toolkit that is meant to simulate the passage of particles through matter.
- It has been developed over the years through collaborative effort of many different institutions and individuals.
- Geant4 has many different applications, including applications in high energy physics, space and radiation, medical.

What is GP-GPU

- General purpose graphic processing unit computing is a re-purposing of graphics hardware
- Allows GPUs to perform computations that would typically be computed on the CPU
- If problems are suitable to mass parallelization

Scope

Purpose

All projects can run in CPU or GPU environment

Able to build GEANT4 with GPU on or off

Why G4ParticleHPVector

Two Implementations

- Run everything on the GPU
- Only select functions run on GPU

Completely on GPU

- The vector is stored exclusively on the GPU
- + Do not have to maintain a copy of the vector on the CPU
- + Do not have to maintain the hashed vector
- + Reduces how much is being copied to the GPU
- All functions are run on the GPU

Intensive Functions on GPU

Performance Results

Performance Discussion

Accuracy

Testing

Implementation 2

- + Only functions that run faster on the GPU are implemented
- + Not forced to run functions that run slowly on GPU
- Will have to maintain two copies of the vector
- More copying the vector to and from the GPU

Performance Summary

- Most functions slower on GPU until ~10,000 entries
- Most *commonly-used* functions significantly slower on GPU
 - Lots of data accesses
- Many problems in vector class not well-suited to parallelism

Performance Results - Times

- Multiplies each point in vector by factor

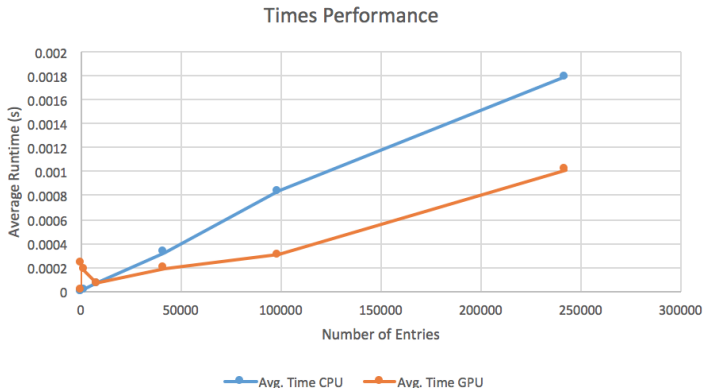


Figure : Runtime vs. Number of Data Points – Times

Performance Discussion

Accuracy

Testing

Summary of Results

Recommendations