

GEANT4-GPU:

User Manual

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

Version 1
April 24, 2016

Contents

1	Revision History	1
2	List of Figures	1
3	Definitions and Acronyms	1
4	Introduction	1
4.1	Purpose of Document	1
4.2	Background	2
4.3	Scope of Document	2
4.4	Document Overview	3
5	Legal Information	3
5.1	Geant4 Software License	3
6	Installation	5
6.1	Supported Operating Systems	5
6.2	Recommended Knowledge	6
6.3	Required Hardware	6
6.4	Required Software	7
6.4.1	Checking GCC Version (RHEL)	7
6.4.2	Checking Clang Version (OS X)	7
6.4.3	CUDA Toolkit	8
6.4.4	CMake	8
6.5	Installation Instructions	8
6.5.1	Installing GEANT4-GPU	9
6.5.2	Installing on McMaster's GPU Servers	11
7	Execution	11
7.1	Enabling/Disabling CUDA	11
7.2	Building the Simulation	12
7.3	Running the Simulation	12
8	Porting Other Geant4 Modules to CUDA	13
8.1	Layout of Source Files	13
8.2	CMake Changes	13
8.3	Interfacing with Existing Code	13
8.4	Naming Conventions	14
9	Troubleshooting	14
9.1	Installation	14
9.2	Running the Simulation	14
9.3	FAQ	14

9.3.1	What is Geant4?	14
9.3.2	How will a GPU improve the performance?	15
9.3.3	Where can I find more information about Geant4?	15
9.3.4	Helpful Pages	15
10	Precautions and Warnings	15

1 Revision History

All major edits to this document will be recorded in the table below.

Table 1: Revision History

Description of Changes	Author	Date
Proper spelling of Geant4,	Stuart	2016-04-24
Fixed links	Matt	2016-03-07
Initial draft of document	Matt, Stuart, Rob, Victor	2016-02-26

2 List of Figures

GEANT4-GPU is a command-line driven program, so only a few figures are included. Commands and results are kept in-line within the text, formatted using a `monospace` typeface.

Table 2: List of Figures

Figure #	Description
1	Results of running original CMake command to generate build files
2	Results of running CMake command to disable CUDA acceleration
3	Visualization of Hadr04 example in Geant4

3 Definitions and Acronyms

4 Introduction

4.1 Purpose of Document

The following user manual corresponds to the McMaster GEANT4-GPU project. The project, under the supervision of Dr. Emil Sekerinski and Dr. Adriaan Buijs, is a modification of the existing Geant4 codebase to leverage GPU computing in certain Geant4 modules. Geant4 is a software toolkit for developing particle simulations.

The main priority of this document is to provide users with all the information needed to install, run, and understand at a high level the GEANT4-GPU project. This includes the license, installation instructions, execution instructions, high-level overview of how CUDA was integrated, and a troubleshooting section.

Table 3: Definitions and Acronyms

Term	Description
Geant4	Open-source software toolkit used to simulate the passage of particles through matter
GEANT4-GPU	Geant4 with some computations running on the GPU
GPU	Graphics processing unit, well-suited to parallel computing tasks
CPU	Computer processing unit, general computer processor well-suited to serial tasks
CUDA	Parallel computing architecture for general purpose programming on GPU, developed by NVIDIA
RHEL	Red Hat Enterprise Linux Server
OS X	Operating system developed by Apple

4.2 Background

There have been several documented attempts in porting parts of Geant4 to the GPU prior to this project. None of these instances approach or discuss the problem in the same modularized fashion.

Alongside niche implementations specifically for medical applications (Bert J, Perez-Ponce H, Lemarchal Y, Gaens M, amongst others) there have also been documented GPU ports of more general Geant4 components (Otto Seiskarim Jukka Kommerim Tapio Niemi).

For background information regarding Geant4, refer to [9.3.1](#).

4.3 Scope of Document

The manual is intended to give basic insight into the nature of GEANT4-GPU and rigorous instructions on how to install and run the project. This document also includes a section for users who wish to integrate CUDA with other aspects of Geant4 in [section 8](#).

This document does not include information on how to use Geant4, or how to develop Geant4 simulations for particular problem domains.

A basic knowledge of programming concepts and command-line tools is assumed, as well as familiarity with Geant4.

4.4 Document Overview

This document first declares the legal information related to the use of the project. Following this are detailed step-by-step instructions on how to install the software. A

section on creating and running simulations using the CUDA functionality introduced in GEANT4-GPU is then included. Following that is a section detailing high-level details about how the CUDA code was developed and integrated, guiding future developers on how to integrate CUDA into other parts of Geant4. A troubleshooting section is included as well as a FAQ section covering many common questions related to the project.

5 Legal Information

This product includes software developed by Members of the Geant4 Collaboration (<http://cern.ch/Geant4>).

5.1 Geant4 Software License

Geant4 Software License
Version 1.0, 28 June 2006

Copyright (c) Copyright Holders of the Geant4 Collaboration,
1994-2006.

See <http://cern.ch/Geant4/license> for details on the
copyright holders. All rights not expressly granted under
this license are reserved.

This software includes voluntary contributions made to Geant4
. See <http://cern.ch/Geant4> for more information on Geant4
.

Installation, use, reproduction, display, modification and
redistribution of this software, with or without
modification, in source and binary forms, are permitted on
a non-exclusive basis. Any exercise of rights by you
under this license is subject to the following conditions:

1. Redistributions of this software, in whole or in part,
with or without modification, must reproduce the above
copyright notice and these license conditions in this
software, the user documentation and any other materials
provided with the redistributed software.
2. The user documentation, if any, included with a
redistribution, must include the following notice:
"This product includes software developed by Members of the
Geant4 Collaboration (<http://cern.ch/Geant4>)."

If that is where third-party acknowledgments normally appear, this acknowledgment must be reproduced in the modified version of this software itself.

3. The names "Geant4" and "The Geant4 toolkit" may not be used to endorse or promote software, or products derived therefrom, except with prior written permission by license@Geant4.org. If this software is redistributed in modified form, the name and reference of the modified version must be clearly distinguishable from that of this software.
4. You are under no obligation to provide anyone with any modifications of this software that you may develop, including but not limited to bug fixes, patches, upgrades or other enhancements or derivatives of the features, functionality or performance of this software. However, if you publish or distribute your modifications without contemporaneously requiring users to enter into a separate written license agreement, then you are deemed to have granted all Members and all Copyright Holders of the Geant4 Collaboration a license to your modifications, including modifications protected by any patent owned by you, under the conditions of this license.
5. You may not include this software in whole or in part in any patent or patent application in respect of any modification of this software developed by you.

6. DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE MEMBERS AND COPYRIGHT HOLDERS OF THE Geant4 COLLABORATION AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, OF SATISFACTORY QUALITY, AND FITNESS FOR A PARTICULAR PURPOSE OR USE ARE DISCLAIMED. THE MEMBERS OF THE Geant4 COLLABORATION AND CONTRIBUTORS MAKE NO REPRESENTATION THAT THE SOFTWARE AND MODIFICATIONS THEREOF, WILL NOT INFRINGE ANY PATENT, COPYRIGHT, TRADE SECRET OR OTHER PROPRIETARY RIGHT.

7. LIMITATION OF LIABILITY

THE MEMBERS AND COPYRIGHT HOLDERS OF THE Geant4 COLLABORATION AND CONTRIBUTORS SHALL HAVE NO LIABILITY FOR DIRECT,

INDIRECT, SPECIAL, INCIDENTAL, CONSEQUENTIAL, EXEMPLARY, OR PUNITIVE DAMAGES OF ANY CHARACTER INCLUDING, WITHOUT LIMITATION, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA OR PROFITS, OR BUSINESS INTERRUPTION, HOWEVER CAUSED AND ON ANY THEORY OF CONTRACT, WARRANTY, TORT (INCLUDING NEGLIGENCE), PRODUCT LIABILITY OR OTHERWISE, ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

8. This license shall terminate with immediate effect and without notice if you fail to comply with any of the terms of this license, or if you institute litigation against any Member or Copyright Holder of the Geant4 Collaboration with regard to this software.

6 Installation

6.1 Supported Operating Systems

GEANT4-GPU has been tested on OS X 10.11 and Red Hat Enterprise Linux Server release 6.7 (Santiago), henceforth called RHEL. Other versions of these operating systems are likely to work, but compatibility is not guaranteed.

To check what version of OS X you have, perform the following actions:

1. Click on the apple icon in the top left-hand corner of your screen.
2. Click on About This Mac.
3. Check to ensure that the version of your OS is 10.11 (El Capitan) (other versions may work, but are not guaranteed).

To check your version of RHEL, perform the following actions:

1. Open a shell.
2. Type in the following command `cat /etc/redhat-release`
3. Check to ensure that the version of RHEL installed is 6.7 (Santiago) (other versions may work, but are not guaranteed).

6.2 Recommended Knowledge

GEANT4-GPU is a scientific product for use in research. There is a large learning curve to using Geant4, and it is recommended that the user has familiarity with the following:

- Comfortable with the Unix command line. This includes navigating the file system, executing make files, and installing software through the command line.
- Some knowledge of programming concepts and C++. Users develop their own simulations in C++ based off of the Geant4 toolkit, and as such some programming experience is recommended.
- Understanding of physics concepts. The user is expected to be knowledgeable in their domain, as Geant4 requires an understanding of the physics processes in the simulation.

6.3 Required Hardware

In order to successfully install and run Geant4 with GPU parallelism, you must first check that the workstation that you are installing on has a NVIDIA GPU with Compute Capability 3.0 or higher.

If the workstation does not have the requisite hardware, GEANT4-GPU may still be installed and executed with GPU computations disabled. In this case however, it is identical to the original Geant4 and it is recommended to use the official version from <http://cern.ch/geant4>.

To check if the GPU on your current workstation supports CUDA with a Compute Capability of 3.0 or higher, perform the following actions:

1. Determine the graphics card that is currently installed (RHEL)
 - (a) Open a shell.
 - (b) Enter the following command: `lspci | grep -i nvidia`
 - (c) You should see output similar to the following:

```
01:00.0 VGA compatible controller: NVIDIA Corporation GK104GL
[Quadro K5000] (rev a1)
```
 - (d) Where [Quadro K5000] is the name of your NVIDIA Graphics Card.
2. Determine the graphics card that is currently installed (OS X)
 - (a) Click on the apple icon in the top left-hand corner of your screen.
 - (b) Click on About This Mac.
 - (c) Click on More Info...
 - (d) Under Contents click Hardware to view it's options.
 - (e) Click on Graphics/Displays.

- (f) You should see the name of your card at the top of the right view-pane.
3. Visit <https://developer.nvidia.com/cuda-gpus> and find your GPU to determine the Compute Capability of your card.
4. Check to ensure that you have a NVIDIA GPU with a Compute Capability of at least 3.0.
5. If you can't find the currently installed GPU on the list, it is not CUDA compatible.

6.4 Required Software

In order to install GEANT4-GPU you must have the following software installed:

- GCC version 4.9 (RHEL) or Clang version 6.0 or later (OS X). Note that McMaster's servers have GCC version 4.1, and an extra command is required to compile, as documented in section 6.5.2.
- CUDA Toolkit v6.5 (other versions may work, however 6.5 is the only one that is tested)
- CMake 2.8 or higher

6.4.1 Checking GCC Version (RHEL)

1. Open a shell.
2. Enter `gcc -v` into the command line.
3. You will see output that looks like: `gcc version 4.9.2 20150212 (Red Hat 4.9.2-6) (GCC)`
4. Ensure that the version number is 4.9

6.4.2 Checking Clang Version (OS X)

1. Open a shell.
2. Enter `clang --version` into the command line.
3. You will see output similar to:
`Apple LLVM version 6.0 (clang-600.0.51) (based on LLVM 3.5svn)`
`Target: x86_64-apple-darwin13.3.0`
`Thread model: posix`
4. Ensure that your version of Clang is at least `clang-600.x.xx`

6.4.3 CUDA Toolkit

If you don't already have the CUDA Toolkit installed, visit <https://developer.nvidia.com/cuda-toolkit-65> and follow the installation instructions for your OS.

After installing, check to see that CUDA is up and running. For either RHEL or OS X:

1. Open a shell.
2. Enter the following command "nvcc -version"
3. Output should be similar to:

```
nvcc:  NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2014 NVIDIA Corporation
Built on Thu Jul 17 21:41:27 CDT 2014
Cuda compilation tools, release 6.5, V6.5.12
```

6.4.4 CMake

If you don't already have CMake (version 2.8 or higher) you can install it from: <https://CMake.org/download/>. After installing CMake, check to see that it is up and running. For either RHEL or OS X 10.11:

1. Open a shell.
2. Enter the following command: `CMake --version`.
3. Output should be similar to:

```
CMake version 3.4.0
CMake suite maintained and supported by Kitware (kitware.com/CMake).
```

6.5 Installation Instructions

Included in the repository at <https://github.com/studouglas/GEANT4-GPU>, there is a README.md file that breaks down the installation steps, including special instructions on how to install this on McMaster's servers. That document is also covered in this section of the user guide.

6.5.1 Installing GEANT4-GPU

1. Clone the repository from <https://github.com/studouglas/GEANT4-GPU>
2. Open a shell

3. Navigate in the shell to the downloaded directory, and run the following command: `mkdir geant4.10.02-build geant4.10.02-install`
4. Navigate to the `geant4.10.02-build` directory you just created.
5. Run the following command to start the build process:

```
cmake -DGEANT4_ENABLE_CUDA=ON \  
-DGEANT4_USE_SYSTEM_EXPAT=OFF \  
-DCMAKE_INSTALL_PREFIX=/path/to/GEANT4-GPU/geant4.10.02-install \  
/path/to/GEANT4-GPU/geant4.10.02
```

- `-DGEANT4_ENABLE_CUDA=ON` – Setting this to `ON` will use your CUDA compiler to compile the CUDA code from the project and link it to the existing C++ code. Setting it to `OFF` (the default) will build Geant4 without any GPU functionality, and without requiring the CUDA compiler or a NVIDIA graphics card.
- `-DGEANT4_USE_SYSTEM_EXPAT=OFF` – This simplifies the installation by using a version of the Expat library that is bundled with Geant4, rather than one that may or may not already be installed on the system.
- `-DCMAKE_INSTALL_PREFIX=...` – Sets the directory to install all the compiled Geant4 libraries, datasets, and examples.

Note that running the above command for the first time can take several hours depending on your internet connection, as it must download large datasets required by Geant4.

6. Once the previous step is completed, all the makefiles required to build GEANT4-GPU are generated, all dependencies are installed, and all that remains is to actually compile Geant4.

In your shell navigate to the `geant4.10.02-build` directory created in step 3 and run the following command: `make install`

This step can take several hours depending on the speed of your computer.

7. GEANT4-GPU is now installed and ready to be used with your simulations. For more information on creating simulations, see section 7.2. If you encountered any errors, refer to section 9.

6.5.2 Installing on McMaster's GPU Servers

If installing GEANT4-GPU on McMaster's GPU servers, the instructions above can be followed with minor differences:

Figure 1: Output of CMake command in step 5 on OS X

```

geant4.10.02-build2 (master)$ cmake -DGEANT4_ENABLE_CUDA=ON -DGEANT4_USE_SYSTEM_EXPAT=OFF -DCMAKE_INSTALL_PREFIX=/Users/stuart/Documents/4th_Year/CS_4ZP6/GEANT4-GPU/geant4.10.02-build2 -bash
-- The C compiler identification is AppleClang 7.3.0.7030029
-- The CXX compiler identification is AppleClang 7.3.0.7030029
-- Check for working C compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++
-- Check for working CXX compiler: /Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Looking for dlfcn.h
-- Looking for dlfcn.h - found
-- Looking for fcntl.h
-- Looking for fcntl.h - found
-- Looking for inttypes.h
-- Looking for inttypes.h - found
-- Looking for memory.h
-- Looking for memory.h - found
-- Looking for stdint.h
-- Looking for stdint.h - found
-- Looking for stdlib.h
-- Looking for stdlib.h - found
-- Looking for strings.h
-- Looking for strings.h - found
-- Looking for string.h
-- Looking for string.h - found
-- Looking for sys/stat.h
-- Looking for sys/stat.h - found
-- Looking for sys/types.h
-- Looking for sys/types.h - found
-- Looking forunistd.h
-- Looking forunistd.h - found
-- Looking for getpagesize
-- Looking for getpagesize - found
-- Looking for bcopy
-- Looking for bcopy - found
-- Looking for memmove
-- Looking for memmove - found
-- Looking for mmap
-- Looking for mmap - found
-- Looking for 4 include files stdlib.h, ..., float.h
-- Looking for 4 include files stdlib.h, ..., float.h - found
-- Check if the system is big endian
-- Searching 16 bit integer
-- Looking for stddef.h
-- Looking for stddef.h - found
-- Check size of unsigned short
-- Check size of unsigned short - done
-- Using unsigned short
-- Check if the system is big endian - little endian
-- Looking for off_t
-- Looking for off_t - not found
-- Looking for size_t
-- Looking for size_t - not found
-- Check size of off64_t
-- Check size of off64_t - failed
-- Looking for fseeko
-- Looking for fseeko - found
-- Looking for unistd.h
-- Looking for unistd.h - found
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - found
-- Found Threads: TRUE
-- Found CUDA: /usr/local/cuda (found version "7.0")
-- CUDA Acceleration ENABLED
-- Reusing dataset G4NDL (4.5)
-- Reusing dataset G4EMLOW (6.48)
-- Reusing dataset PhotonEvaporation (3.2)
-- Reusing dataset RadioactiveDecay (4.3)
-- Reusing dataset G4NEUTRONXS (1.4)
-- Reusing dataset G4PII (1.3)
-- Reusing dataset RealSurface (1.0)
-- Reusing dataset G4SAIDDATA (1.1)
-- Reusing dataset G4ABLA (3.0)
-- Reusing dataset G4ENSDFSTATE (1.2)
-- The following Geant4 features are enabled:
GEANT4_BUILD_CXXSTD: Compiling against C++ Standard '11'

-- Configuring done
-- Generating done
-- Build files have been written to: /Users/stuart/Documents/4th_Year/CS_4ZP6/GEANT4-GPU/geant4.10.02-build2
geant4.10.02-build2 (master)$

```

- The default gcc compiler on the servers is too old to build Geant4, so add the following command to your `.bash_profile` to use a newer version:
`. /opt/rh/devtoolset-3/enable`
- In step 5, add the following flag to the CMake command:
`-DCUDA_HOST_COMPILER=/usr/bin/g++`. This compiles CUDA with the older version of gcc that's installed on the server, as the latest version is not supported.

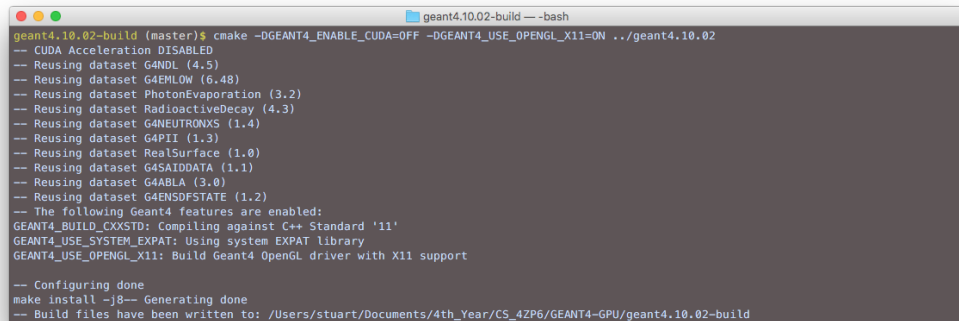
7 Execution

Geant4 is not an executable program in the traditional sense. It is instead a large set of libraries, designed to work together and that give you, the user, a framework to develop simulations for your work. The development of such simulations is not within the scope of this manual, but it is well-documented with a thorough guide available at <http://Geant4.web.cern.ch/Geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/index.html>.

This section will instead outline how one would go about using (or not using) CUDA computations with an existing simulation. The `Hadr04` example that comes with Geant4 will be used as the simulation to demonstrate this.

7.1 Enabling/Disabling CUDA

Figure 2: Results of CMake command to disable CUDA



```

geant4.10.02-build (master)$ cmake -DGEANT4_ENABLE_CUDA=OFF -DGEANT4_USE_OPENGL_X11=ON ../geant4.10.02
-- CUDA Acceleration DISABLED
-- Reusing dataset G4NDL (4.5)
-- Reusing dataset G4EMLOW (6.48)
-- Reusing dataset PhotonEvaporation (3.2)
-- Reusing dataset RadioactiveDecay (4.3)
-- Reusing dataset G4NEUTRONXS (1.4)
-- Reusing dataset G4PII (1.3)
-- Reusing dataset RealSurface (1.0)
-- Reusing dataset G4SAIDDATA (1.1)
-- Reusing dataset GAABLA (3.0)
-- Reusing dataset G4ENSDFSTATE (1.2)
-- The following Geant4 features are enabled:
GEANT4_BUILD_CXXSTD: Compiling against C++ Standard '11'
GEANT4_USE_SYSTEM_EXPAT: Using system EXPAT library
GEANT4_USE_OPENGL_X11: Build Geant4 OpenGL driver with X11 support

-- Configuring done
make install -j8-- Generating done
-- Build files have been written to: /Users/stuart/Documents/4th_Year/CS_42P6/GEANT4-GPU/geant4.10.02-build

```

The installation guide details how to build the Geant4 libraries that will be used by your simulation. The CMake command from step 5 includes a flag that determines whether or not CUDA will be used – `-DGEANT4_ENABLE_CUDA=ON`. Changing this value to `OFF` will cause Geant4 to build without any CUDA features, and without requiring `nvcc`, the CUDA compiler. After step 5 finishes, users will have to rebuild Geant4 by

executing the `make install` command from their `geant4.10.02-build` directory. To re-enable CUDA, simply change the value to `ON` and rebuild again.

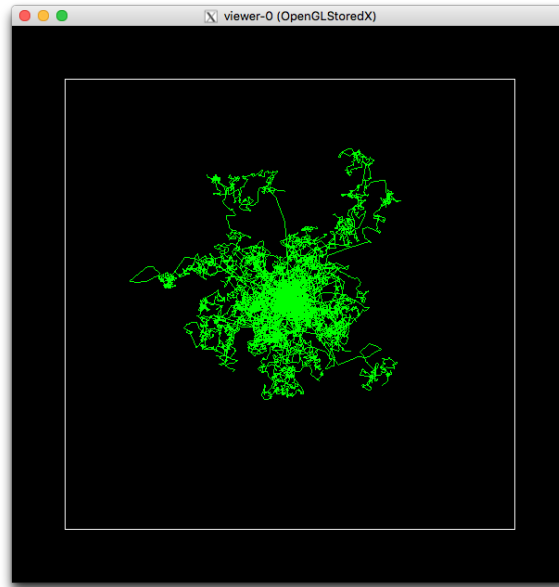
7.2 Building the Simulation

After Geant4 has been built, building the simulation is very straightforward. First, navigate to your simulation's root folder (`geant4.10.02/examples/hadronic/Hadr04` in our case) in your terminal. Then, create a new directory `bin`, so your `Hadr04` folder now has two folders, `bin` and `src`. Go into the `bin` folder and run `cmake ../src` followed by `make`. That's it!

7.3 Running the Simulation

There are two main methods of running a Geant4 simulation. The first is to simply run the executable, in our case by running `./Hadr04` in the `Hadr04` directory. This launches an interactive Geant4 command prompt, and you can manipulate all aspects of the simulation from within it, including the types of particles, their number, and their energies. This also allows you to visualize the simulation using the `vis` commands.

Figure 3: Hadr04 Example with Visualization



The alternative is to create a macro file that contains a sequence of commands that would be manually given to the interactive prompt, making it much easier to run a given simulation more than once and without requiring user interaction. Both methods print the results of the simulation out to the terminal window, as well as the running time of the simulation. If you wish to save the results to a file, that can easily be

done via redirection in Unix. For example, if you wish to save the results of your simulation using `myMacro.mac` to `results.txt`, simply run `./Hadr04 myMacro.mac > results.txt`.

8 Porting Other Geant4 Modules to CUDA

There is currently one class in GEANT4-GPU that uses CUDA – `G4ParticleHPVector`. This class was originally chosen as it was thought to be well-suited to parallelism. Although certain functions within the class are, the majority of the time spent within the class is not in these functions, the main reason for the performance degradations when CUDA is enabled.

There is indeed the possibility that there are other classes within Geant4 that are better suited to parallelism. This section will outline the general methodology we used to port `G4ParticleHPVector` with the hopes that it may be of use to future developers or users.

8.1 Layout of Source Files

An important consideration during development was to keep the CUDA code separate from the main Geant4 codebase as much as possible. A directory was created named `cuda` in `geant4.10.02/source/externals` to contain all CUDA source code. The `cuda` folder contains two folders, `include` and `src`, as well as the `CMakeLists.txt` file with the CUDA compilation directives. All header files are in `include`, and the CUDA source files are in `src`.

8.2 CMake Changes

Geant4 uses CMake as its build system, and minor modifications need to be made to integrate with CUDA. This consists of adding a variable at the top level `CMakeLists` file to enable or disable CUDA, and creating a `Geant4_Add_Library_Cuda` macro to `Geant4MacroLibraryTargets`. This makes use of the `cuda_add_library` function included in CMake. More details can be found in the project's detailed design document.

8.3 Interfacing with Existing Code

We used compiler directives within the default `G4ParticleHPVector.cc` file to make use of the CUDA code or to use the original implementation based on the flag passed to Geant4 during the build phase. If the flag is true, an object of type `G4ParticleHPVector_CUDA` is initialized and all function calls become one line calls to the corresponding function on the `G4ParticleHPVector_CUDA` object. For example, the `GetY` function looks like the following:


```

G4double GetY(int i) {
    #if GEANT4_ENABLE_CUDA
        return cudaVector->GetY();
    #else
        < existing code from G4ParticleHPVector.cc >
    #endif
}

```

8.4 Naming Conventions

It is important to use naming conventions to ensure clarity within a project that contains many possible ambiguities. The two main conventions used were adding a “_CUDA” suffix to all source files within the `cuda` directory as well as to the class name of any classes implemented in CUDA. In addition to this, within all CUDA files there are two main types of pointer – those to GPU memory and those to CPU memory. To distinguish between these, all pointers to main memory are prepended with “h_” (for host) and pointers to GPU memory with a “d_” (for device).

9 Troubleshooting

9.1 Installation

- Ensure path names are correct in step 5 of installation
- The paths in step 5 of installing Geant4 paths must be absolute, relative paths will not work
- Spaces in path names may cause issues on some systems

9.2 Running the Simulation

- Ensure that your graphics card meets the requirements detailed in section 6.3

9.3 FAQ

9.3.1 What is Geant4?

Many physics researchers use Geant4 to learn about how particles interact with a specific environment. It is a toolkit (i.e. a set of libraries) that uses the Monte Carlo model, meaning each particle’s properties are calculated independently according to certain probabilities. Users develop simulations based on these libraries.

9.3.2 How will a GPU improve the performance?

GPU's contain a large amount of cores that can perform calculations much more quickly than a CPU if the problem is well-suited to parallelization. Geant4 runs relatively simple calculations on millions of particles, and each particle is completely independent of the others. This is exactly that sort of well-suited problem, and stands to see large performance gains.

9.3.3 Where can I find more information about Geant4?

For more information, check out the official website at <http://Geant4.web.cern.ch/Geant4/>.

9.3.4 Helpful Pages

- Download page for Geant4 source code: <http://Geant4.web.cern.ch/Geant4/support/download.shtml>
- Getting Started guide: <http://Geant4.web.cern.ch/Geant4/support/gettingstarted.shtml>
- Installation Guide for Geant4 (does not include CUDA): <http://Geant4.web.cern.ch/Geant4/UserDocumentation/UsersGuides/InstallationGuide/html/index.html>

10 Precautions and Warnings

By default when running a simulation all output is displayed in the Terminal window. This means that closing the terminal window after a simulation completes will lose all the results. To prevent this, it is recommended to redirect the output to a file, as described in section 7.3.