

CUDA Education

Parallel Programming Education Materials

Whether you're looking for presentation materials or CUDA code samples for use in education self-learning purposes, this is the place to search!

Please keep checking back as new materials will be posted as they become available. We recommended you subscribe to the following e-mail list to be kept informed of updates and any Parallel Programming education events held in the future.

[SIGN-UP TODAY \(/content/cuda-education-information-sign\)](#)

To report issues with the materials linked below, make requests for materials, or discuss their use, please participate on the Parallel Programming Education forum (<https://devtalk.nvidia.com/default/board/125/parallel-programming-education/>) on developer forums devtalk.nvidia.com.



Presentations

Traditionally, presentation slides are distributed and downloaded as PDF formatted files. While this may allow for a greater number of systems to more easily view the slides, it prevents someone from using and building on the existing slides. The goal for the presentations in this section is to allow educators to fully utilize and modify the content to fit within their curriculum.

The only requirement for using these presentations is that you give recognition to the original author as listed below and in most of the files themselves.

General Purpose

Introduction to CUDA Platform

Author: Will Ramey – NVIDIA Corporation

Description: This deck covers the basics of what makes up the CUDA Platform. No longer just a C compiler, CUDA has changed greatly since its inception and is now the platform for parallel computing on NVIDIA GPUs. Use this presentation to help educate on the different areas of the CUDA platform and different approaches for programming GPUs

Downloads:

- Presentation here

(http://developer.download.nvidia.com/compute/developertrainingmaterials/presentations/general/Introduction_to_CUDA_Platform.pptx)

Why GPU Computing

Author: Mark Ebersole – NVIDIA Corporation

Description: One of the most important parts of GPU computing is helping to convince people why it's so important. This deck is intended to help present the data and educate people on the GPU computing revolution.

Downloads:

- Presentation here

(http://developer.download.nvidia.com/compute/developertrainingmaterials/presentations/general/Why_GPU_Computing.pptx)

CUDA Languages

Introduction to CUDA C

Author: Mark Harris ([http://en.wikipedia.org/wiki/Mark_Harris_\(programmer\)](http://en.wikipedia.org/wiki/Mark_Harris_(programmer)))– NVIDIA Corporation

Description: Starting with a background in C or C++, this deck covers everything you need to know in order to start programming in CUDA C. Beginning with a "Hello, World" CUDA C program, explore parallel programming with CUDA through a number of code examples. Examine more deeply the various APIs available to CUDA applications and learn the best (and worst) ways in which to employ them in applications.

Downloads:

- Presentation here

(http://developer.download.nvidia.com/compute/developertrainingmaterials/presentations/cuda_language/Introduction_to_CUDA_C.pptx)

- Presentation of these slides by Justin Luitjens from NVIDIA Corporation recorded here (<http://nvidia.fullviewmedia.com/gtc2012/0514-A5-S0624.html>)

References:

- See VectorAdd code sample here

- See 1DStencil code sample here

- See a version of the Hello World code sample here

Introduction to Thrust Parallel Algorithms Library

Author: Thomas Bradley – NVIDIA Corporation

Description: Thrust is a parallel algorithms library which resembles the C++ Standard Template Library (STL). Thrust's high-level interface greatly enhances developer productivity while enabling performance portability between GPUs and multicore CPUs. Interoperability with established technologies (such as CUDA, TBB and OpenMP) facilitates integration with existing software. This presentation walks through the library's main features and explains how developers can build high-performance applications rapidly with Thrust.

Downloads:

- Presentation here

(http://developer.download.nvidia.com/compute/developertrainingmaterials/presentations/cuda_language/Introduction_to_Thrust.pptx)

- Presentation of these slides by Nathan Bell and Julien Demouth from NVIDIA Corporation recorded here (<http://nvidia.fullviewmedia.com/gtc2012/0515-A3-S0602.html>)

Introduction to Libraries

Author: Mark Harris ([http://en.wikipedia.org/wiki/Mark_Harris_\(programmer\)](http://en.wikipedia.org/wiki/Mark_Harris_(programmer)))– NVIDIA Corporation

Description: A very brief presentation to be used as an introduction to CUDA libraries.

Downloads:

- Presentation here

(http://developer.download.nvidia.com/compute/developertrainingmaterials/presentations/cuda_language/Introduction_to_Libraries.pptx)

Parallel Programming

Martel Reduce Scan Sort

Author: John Owens (<http://www.ece.ucdavis.edu/~jowens/>)– University of California, Davis

Description: This deck covers parallel Reduction, Scan, Sort, and Merge on a SPMD + SIMD architecture, of which an NVIDIA GPU is used. Different versions of each are covered, as well as some in-depth analysis on the efficiencies and performance of them.

Downloads:

- Presentation here

(http://developer.download.nvidia.com/compute/developertrainingmaterials/presentations/parallel_programming/martel-reduce-scan-sort.ppt)

OpenACC

Introduction to OpenACC

Author: Mark Harris ([http://en.wikipedia.org/wiki/Mark_Harris_\(programmer\)](http://en.wikipedia.org/wiki/Mark_Harris_(programmer)))– NVIDIA Corporation

Description: OpenACC is an open programming standard for parallel computing on accelerators (including GPUs) using directives. It is designed to make the transformative power of heterogeneous computing systems available to the developer quickly and easily. This presentation will help explain how to add simple directives to code to expose parallelism to the compiler, allowing it to efficiently map computation onto an accelerator automatically.

Downloads:

- Presentation here

(http://developer.download.nvidia.com/compute/developertrainingmaterials/presentations/openacc/Introduction_To_OpenACC.pptx)

- Presentation of these slides by various presenters recorded here (<http://www.gputechconf.com/gtcnew/on-demand-gtc.php?searchByKeyword=S0517&searchItems=&sessionTopic=&sessionEvent=2&sessionYear=2012&sessionFormat=&submit=&select=+>)

References:

- See the Laplace2D OpenACC code sample here

Debugging

No content posted yet. If you have a material you are interested in sharing, please post in the Parallel Programming forum (<http://devtalk.nvidia.com/default/board/125/parallel-programming-education/>).

Optimization

No content posted yet. If you have a material you are interested in sharing, please post in the Parallel Programming forum (<http://devtalk.nvidia.com/default/board/125/parallel-programming-education/>).

Tools and Utilities

No content posted yet. If you have a material you are interested in sharing, please post in the Parallel Programming forum (<http://devtalk.nvidia.com/default/board/125/parallel-programming-education/>).

Code Samples for Education

There are many CUDA code samples available online, but not many of them are useful for teaching specific concepts in an easy to consume and concise way. The goal for these code samples is to provide a well-documented and simple set of files for teaching a wide array of parallel programming concepts using CUDA.

The only requirement for using these code samples in your courses is to give recognition to the original author listed below and in most of the files themselves.

Each sample listed below includes a README file describes the code in detail. It will also give teaching flow recommendations, compile & run instructions, as well as any references or files required.

Finally, the samples contained in the CUDA SDK are a fantastic resource for demonstrating how different methods and techniques are written. You can see the samples and their source at here (<http://docs.nvidia.com/cuda/cuda-samples/index.html>).

CUDA C

· Hello World example

Author: Mark Ebersole – NVIDIA Corporation

Description: A simple version of a parallel CUDA “Hello World!”

Downloads:

- Zip file here (http://developer.download.nvidia.com/compute/developertrainingmaterials/samples/cuda_c/HelloWorld.zip)

· VectorAdd example

Description: A CUDA C program which uses a GPU kernel to add two vectors together. All the memory management on the GPU is done using the runtime API.

Downloads:

- Zip file here (http://developer.download.nvidia.com/compute/developertrainingmaterials/samples/cuda_c/VectorAdd.zip)

· 1DStencil example

Description: A CUDA C program which calculates a 1DStencil, making use of shared memory and synchronized threads to achieve better performance.

Downloads:

- Zip file here (http://developer.download.nvidia.com/compute/developertrainingmaterials/samples/cuda_c/1DStencil.zip)

References:

- Also includes a presentation describing the algorithm.

· Jacobi Optimization

Author: Pradeep Kumar Gupta - NVIDIA Corporation

Description: This code performs Point Jacobi Iterative method on CPU (sequential code) and on NVIDIA GPU for 2D data. This code is developed for reference and demonstrates the speedup using CUDA and can be optimized further.

Downloads:

- Zip file here (http://developer.download.nvidia.com/compute/developertrainingmaterials/samples/cuda_c/Jacobi_Optimization.zip)

References:

- Also includes a presentation describing the algorithm.

CUDA Python

· Mandelbrot iPython Notebook Example

Author: Mark Harris ([http://en.wikipedia.org/wiki/Mark_Harris_\(programmer\)](http://en.wikipedia.org/wiki/Mark_Harris_(programmer)))- NVIDIA Corporation

Description: This example starts with a single-threaded, interpreted python mandelbrot algorithm and progresses to a CUDA accelerated version which will run incredibly fast on a modern GPU. While not immediately available as a hands-on lab, the implementation in a ipython notebook (<http://ipython.org/notebook.html>) makes it easily convertible to hands-on format. Please note that this code requires a CUDA Python enabled compiler, such as NumbaPro, which is part of the Anaconda Accelerate (<https://store.continuum.io/cshop/accelerate/>) package from Continuum Analytics.

Downloads:

- Github repository here (https://github.com/harrism/numbapro_examples)
- Read-only notebook view here (<http://nbviewer.ipython.org/f5707335f40af9463c43>)

Monte Carlo iPython Notebook Example

Author: Siu Kwan Lam - Continuum Analytics

Description: This example starts with a single-threaded, interpreted python monte carlo algorithm and progresses to a CUDA accelerated version which will run incredibly fast on a modern GPU. While not immediately available as a hands-on lab, the implementation in a ipython notebook (<http://ipython.org/notebook.html>) makes it easily convertible to hands-on format. Please note that this code requires a CUDA Python enabled compiler, such as NumbaPro, which is part of the Anaconda Accelerate (<https://store.continuum.io/cshop/accelerate/>) package from Continuum Analytics.

Downloads:

- Github repository here (https://github.com/ContinuumIO/numbapro-examples/tree/master/monte_car)
- Read-only notebook view here (<http://nbviewer.ipython.org/835a8ca39ced77fe751d>)

OpenACC

Laplace2D example

Description: A simple Jacobi iteration useful for teaching both the kernels (or parallel) and data directives. There is some high-level timing code built-in to the program, but you may wish to also make use of a profiler when using this example.

Downloads:

- Zip file here (<http://developer.download.nvidia.com/compute/developertrainingmaterials/samples/openacc/Laplace2D.zip>)

OpenACC + OpenGL Interoperation example

Author: Peter Messmer - NVIDIA Corporation

Description: This sample demonstrates the use OpenGL from within an OpenACC application. The chosen approach allows visualizing the data resident on the GPU without transferring it back to the host.

Downloads:

- Zip file here (<http://developer.download.nvidia.com/compute/developertrainingmaterials/samples/openacc/openAccGL.zip>)

CUDA ZONE

[/cuda-zone]

[/CUDA-ZONE]

GAMEWORKS

[/gameworks]

[/GAMEWORKS]

PROFESSIONAL GRAPHICS

[\(/professional-graphics\)](#)
[\(/PROFESSIONAL-GRAPHICS\)](#)

GET STARTED

[About CUDA \(/about-cuda\)](#)
[Parallel Computing \(/get-started-parallel-computing\)](#)
[CUDA Toolkit \(/cuda-toolkit\)](#)
[CUDACasts \(http://www.youtube.com/playlist?list=PL5B692fm6--vScfBaxgY89lRWFzDt0Khm\)](http://www.youtube.com/playlist?list=PL5B692fm6--vScfBaxgY89lRWFzDt0Khm)

LEARN MORE

[Training and Courseware \(/cuda-education-training\)](#)
[Tools and Ecosystem \(/cuda-tools-ecosystem\)](#)
[Key Technologies \(/key-technologies\)](#)
[Documentation \(http://docs.nvidia.com/cuda/index.html\)](http://docs.nvidia.com/cuda/index.html)

GET INVOLVED

[Forums \(https://devtalk.nvidia.com/\)](https://devtalk.nvidia.com/)
[Parallel Forall Blog \(https://devblogs.nvidia.com/parallelforall/\)](https://devblogs.nvidia.com/parallelforall/)
[Developer Program \(/registered-developer-program\)](#)
[Contact Us \(/contact\)](#)