

Cuda for Physicists

David Poliakoff

Jitu Das

Jonathan Gluck

Advisors: Dr. Randall Hall and Dr. Mark
Jarrell

What is CUDA?

- Software interface to NVIDIA graphics cards.
- Programming for graphics cards can lead to incredible speed increases using (relatively) cheap hardware.
- Budget supercomputing

What CUDA isn't

- Accessible for non-programmers

- By default, C style memory allocation
- Lacks some fairly basic functions (Random Number Generator)

- Without “quirks”

- Memory has a nasty habit of disappearing between functions
- Generally, loads of things you don't need to worry about in serial processing come into play

What I'm doing

- Often along with Jonathan Gluck and Jitu Das, I'm looking to help scientists (primarily physicists, but others as well) implement code on CUDA without too much fuss
- Hopefully we'll also get something more generally useful (a useful library of code for general use)

Challenges

- Simplify thread management/CUDA calls
 - Thread – CUDA Threads can be thought of as series of instructions which must run concurrently
- “common_functions.h” is impossible
 - Data retention between CUDA calls
 - Breaks a LOT of software engineering ideas

Desirable Outcome

- Code snippets (not functions) that are easy to throw within a given CUDA function

- Snippets are text to copy between functions, functions are compilable
- Need to be simple and efficient
- Examples
 - “Random” number generation
 - Basic Linear Algebra Subprograms (BLAS)

Main Products

- Dr. Hall and Dr. Jarrell are giving us some of the simulations they run
- Metropolis-Hastings algorithm
 - Jitu and Jonathan will give more details