Easy, Effective, Efficient: GPU Programming in Python with PyOpenCL and PyCUDA

Andreas Klöckner

Courant Institute of Mathematical Sciences
New York University

PASI: The Challenge of Massive Parallelism
Lab · January 7/8, 2011

A Monte-Carlo Simulation

In this lab, we will consider a Monte Carlo simulation.

For each sample, do:

1. Generate a vector x of random numbers on the CPU.
2. Transfer x to the GPU.
3. Compute $y = Ax + b$, for some matrix A and vector b.
4. Transfer y back to the CPU.

When finished, plot histogram of distribution of 2-norms $\|y\|$.

Problem

Make this code go as fast as you can.

Overarching Goal

Here are some things to try, in order of increasing difficulty:

1. Insert (event-based) fine-grained timing code.
2. Overlap Host$\leftrightarrow$GPU transfers with computation.
   - Turn off profiling, use page-locked memory for actual overlap (At least on Nvidia)
3. Compute 2-norms on the GPU.
4. Generate random numbers on GPU.
5. Compute Ax for multiple x alongside each other
   - Perhaps load (parts of) x into local memory.

Do 1, 2 and 3, after that pick one that looks like it'll be fun.

`http://tiker.net/tmp/pasi-lab.pdf`

Overarching Goal

Here are some things to try, in order of increasing difficulty:

1. Insert (event-based) fine-grained timing code.
2. Overlap Host↔GPU transfers with computation.
   - Turn off profiling, use page-locked memory for actual overlap (At least on Nvidia)
3. Compute 2-norms
4. Generate random
5. Compute Ax for r
   - Perhaps load

Do 1, 2 and 3, after th

Now:

- Look at the code
- Try running it, and
- Start on task 1 (add timing).

We will reconvene in 15–20 minutes for some discussion.

Login Instructions

To get to your work environment, do the following:

1. `ssh pasiNN@gpu.prograpе.jp`
2. `wget http://tiker.net/tmp/pasi-lab.tar.gz`
3. `tar xvfz pasi-lab.tar.gz`
4. `cd student-dir/monte-carlo`
5. `python pasi-lab.py`
6. `wget http://tiker.net/tmp/pasi-lab-1-instrumented.py`

`http://tiker.net/tmp/pasi-lab.pdf`

Questions?

?

Any questions about the code?

http://tiker.net/tmp/pasi-lab.pdf

Making performance guesstimates

With your first look complete, let's try and answer these questions:

- Where is the most time being spent?
- Is the matrix-vector code compute- or memory-bound?
- Which code change will give the greatest performance win?

http://tiker.net/tmp/pasi-lab.pdf