

Scripting GPUs with PyOpenCL

Hands-on Session

Andreas Klöckner

Division of Applied Mathematics
Brown University

October 8, 2009

OpenCL Demo Machines

Getting Started

- 1 Everybody gets a two-digit number: NN
(pad with zeros at the front, e.g. 05)
- 2 `ssh fcj09NN@haamster.rice.edu`
(haamster is the login node—don't compute here.)
- 3 Note down your password.
- 4 If your number is odd: `ssh teramite` (6 GPUs)
If your number is even: `ssh slate` (5 GPUs)
- 5 `export CUDA_DEVICE=(NN mod #GPUs)`

Please write down all of this information *now*.

GPU demo machines courtesy of Tim Warburton, Rice University.



BROWN

OpenCL Coding Practice

Hands-on Exercise

- 1** Write a program that efficiently multiplies matrices whose row and column count are multiples of 16.
Start from `hands-on/start.py`.
(Solutions exist in the `pyopencl-examples` directory.
But don't look.)
- 2** Implement benchmarking logic.
- 3** How fast is your code? What do you expect to be the limiting factor in performance? What is the device's peak performance? Are you getting close? Why not? Can you improve your code?

OpenCL Coding Practice

Once you're "done" . . .

- 1 Pick a problem from your field.
- 2 Start hacking.
- 3 Profit! :-)

PyOpenCL documentation:

<http://documen.tician.de/pyopenc1>

OpenCL spec: (for device language)

<http://khronos.org/registry/cl>

PyCUDA documentation:

<http://documen.tician.de/pycuda>

