Bridges GPU and Large Memory Nodes

John Urbanic
Parallel Computing Scientist
Pittsburgh Supercomputing Center
3 Ways to Accelerate Applications on GPUs

Applications

Libraries
“Drop-in” Acceleration

OpenACC Directives
Easily Accelerate Applications

Programming Languages (CUDA, OpenCL)
Maximum Flexibility
NVIDIA cuBLAS
NVIDIA cuRAND
NVIDIA cuSPARSE
NVIDIA NPP

GPU VSIPL
Vector Signal Image Processing

CULA tools
GPU Accelerated Linear Algebra

MAGMA
Matrix Algebra on GPU and Multicore

NVIDIA cuFFT

ROGUE WAVE SOFTWARE
IMSL Library

libjacket
Building-block Algorithms for CUDA

CUSP
Sparse Linear Algebra

Thrust
C++ STL Features for CUDA

GPU Accelerated Libraries
“Drop-in” Acceleration for Your Applications
OpenACC Champion Case

4x Faster

**Jaguar**  **Titan**

42 days  10 days

---

Modified <1%

Lines of Code

---

15 PF! One of fastest simulations ever!

Design alternative fuels with up to 50% higher efficiency

**S3D: Fuel Combustion**
GPUs

Everyone loves GPUs these days, and we will have the leading edge cards: 2 K80’s per node, with Pascal soon to arrive (GTC next week!).

And they will be embedded in a great hardware infrastructure: CPU hosts, filesystems, network. If you have a developed application you are ready to go.

But it is the software and development environment that matters most if you are building something:

- CUDA
- CUDA Libs
- OpenACC*
- CUDA Toolkit
- GPU enabled libraries

*You probably want to use the PGI compiler here, but perhaps this is a wise choice for GPU development in general…
This is for an OpenACC Mandlebrot set image generation code from NVIDIA. You can grab it at

https://github.com/NVIDIA-OpenACC-Course/nvidia-openacc-course-sources
Large Memory

We know how to enable large memory usage. For a long time we had the world’s largest memory machine here at PSC, Blacklight. It had 2 16TB partitions. So having 3TB and 12TB nodes is something we appreciate.
There are lots of applications that aren’t amenable to distributed memory, but require more RAM than you have on any available workstation.

- Genomics
- Quantum Chemistry
- In memory databases
- Graph Analytics (Neo4j)
- Machine Learning on large training sets
- Workstation oriented packages: Matlab, more to come…
Large Memory Programming

The programming environment supports the well-established shared memory programming models. You (or your package) are most likely to be using:

- OpenMP (easier than you probably think)
- Java