Accelerators are the Future of HPC

- HPC can't support its own designs
  - commodity parts, software as well as hardware
- Accelerators allow break from ISA compatibility
- Accelerators allow strong scaling
- GPUs are the game today
  - designs are essentially free
- You must program to the new model
- Downside
  - future of commodity may be in mobile
  - commodity cpus and accelerators may not solve HPC problems

Programming Accelerators

- Goals
  - productivity
  - performance
  - portability
- Will it run fast on tomorrow's accelerators?

Options

- Low level
  - OpenCL, CUDA
  - full control
  - low on productivity, performance portability
  - high on performance
  - language is portable, even if programs are not
Options

- Libraries
  - Magma, etc.
  - programming to the library
  - essentially a limited-vocabulary language
  - high on portability, productivity, performance
    - if your program fits the vocabulary

Options

- Class library
  - TBB, Ct, (Rapidmind), Thrust
  - A type system and implementation
  - Advantage: some information instantiated at compile time
  - Other advantages / disadvantages are the same as library approach

Options

- High level, PGI Accelerator model, (eventually OpenMP)
  - High on productivity, portability
  - Performance is improving over time
  - Open question: how portable is the model?

How to Reach a Petaflop

- $10^6$ = megaflop
- $10^9$ = gigaflop
- $10^{12}$ = teraflop
- $10^{15}$ = petaflop
  - Jaguar
    - 16,688 dual-socket six-core nodes
      - 2.6GHz, 4-8 GFlops/core, 224,256 cores
    - $(\cdot 224 \times 10^9$ cores) x (2.6 x $10^9$ GHz) x (4 results)
      - $2.32 \times 10^{15}$ results/cycle = 2.32 Petaflops (double precision)
    - Top500 Rmax = 1.759 PFlops, Rpeak = 2.331 PFlops
How to Reach an Exaflop

- $10^6 = \text{megaflop}$
- $10^9 = \text{gigaflop}$
- $10^{12} = \text{teraflop}$
- $10^{15} = \text{petaflop}$
- $10^{18} = \text{exaflop} = 10^9 \times 10^9$
  - one billion gigaflop cores = one exaflop (MPI$$)$$^*$
    - 1 million quad-socket 256-core nodes at 1GHz
    - 50X nodes, 2X sockets/node, 40X cores/socket relative to Jaguar
    - at 4 results/GHz, reduce by 1/4, higher clock reduces as well
  - one million teraflop cores = one exaflop
    - 1,000 ops / cycle (1GHz)

How to Reach an Exaflop

- Maybe O(100,000) 10-teraflop nodes
  - wide SIMD, multitreading, latency tolerant
  - 1GHz clock = 10,000 operations/cycle (5,000 mul+add)

Jaguar

- Proposed

<table>
<thead>
<tr>
<th>Jaguar</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>224,256 cores</td>
<td>O(100,000) units</td>
</tr>
<tr>
<td>37,376 sockets</td>
<td>100,000 sockets</td>
</tr>
<tr>
<td>18,688 nodes</td>
<td>O(100,000) nodes</td>
</tr>
<tr>
<td>2.5GHz clock</td>
<td>1GHz clock</td>
</tr>
<tr>
<td>4-8 GFlops/core</td>
<td>O(1,000) GFlops/unit</td>
</tr>
<tr>
<td>24-48 GFlops/socket</td>
<td>1,000 GFlops/socket</td>
</tr>
<tr>
<td>48-96 GFlops/node</td>
<td>O(1,000) GFlops/node</td>
</tr>
</tbody>
</table>

Additional Information

- PGI Accelerator Programming Model
  - x86 + NVIDIA
  - PGI Fortran and C
  - Linux, Windows, OSX
  - [www.pgroup.com/accelerate](http://www.pgroup.com/accelerate) for documentation, FAQ, articles

- PGI CUDA Fortran
  - x86 + NVIDIA
  - PGI Fortran
  - Linux, Windows, OSX
  - [www.pgroup.com/cudafortran](http://www.pgroup.com/cudafortran) for documentation, FAQ, articles

- Common Compiler Feedback Format (CCFF)
  - integrated into all PGI compilers and pgprof
  - [www.pgroup.com/CCFF](http://www.pgroup.com/CCFF) for additional information

Copyright Notice

© Contents copyright 2009-2011, The Portland Group, Inc. This material may not be reproduced in any manner without the expressed written permission of The Portland Group.