PROGRAMMING GPU SUPERCOMPUTERS TEN YEARS FROM NOW

Michael Wolfe, March 26, 2018
PGI — THE NVIDIA HPC SDK

Fortran, C++, C Compilers
- Optimizing, SIMD Vectorizing, OpenMP

Accelerated Computing Features
- OpenACC Directives, CUDA Fortran

Multi-Platform Solution
- X86-64 and OpenPOWER Multicore CPUs
- NVIDIA Tesla GPUs
- Supported on Linux, macOS, Windows

MPI/OpenMP/OpenACC Tools
- Debugger
- Performance Profiler
- Interoperable with DDT, TotalView
NVIDIA CUDA
Compute Unified
Device Architecture

Programming Guide

Version 1.0

The SC07 Tutorials Program gives attendees the opportunity to explore a wide variety of important topics related to high performance computing, networking, and storage.

<table>
<thead>
<tr>
<th>Time</th>
<th>Session</th>
<th>Title</th>
<th>Speakers</th>
<th>Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>8:30AM - 5:00PM</td>
<td>S05</td>
<td>High Performance Computing on GPUs with CUDA</td>
<td>Massimiliano Fatica, David P. Luebke, Ian A. Buck, John D. Owens, Mark J. Harris, John E. Stone, James C. Phillips, Bernard Deschizeaux</td>
<td>A3</td>
</tr>
<tr>
<td>8:30AM - 5:00PM</td>
<td>S06</td>
<td>Introduction to Globus</td>
<td>Jennifer M. Schopf, Ben Clifford, Ravi</td>
<td>A20</td>
</tr>
</tbody>
</table>
GPU COMPUTING

- System Memory
- GPU Memory

PCI / NVLink

- Data management
- Compute management
- Compute optimization
- Asynchronous operation
PROGRAMMING GPUS TODAY

- Libraries
- Data management
- Compute management
- Compute optimization
- Asynchronous operation

- NVIDIA CUDA
- OpenACC
- Python
- OpenCL
- OpenMP
- MATLAB

PGI
THE FUTURE OF GPU DATA MANAGEMENT

Michael Wolfe, May 9, 2017
GPU COMPUTING

Unified Memory

data management
compute management
compute optimization
asynchronous operation
GPU PROGRAMMING
Focusing on HPC In Ten Years

This presentation contains predictions and forward-looking statements
All opinions are those of the presenter
GPU PROGRAMMING
Focusing on HPC In Ten Years

C++    Fortran
C++

```cpp
for_each_n(par, 0, n, [&] (int i) {
    y[i] += a*x[i];
});
```

Fortran

```fortran
do concurrent (i = 1:n)
    y(i) = y(i) + a*x(i)
endo
```

Focusing on HPC In Ten Years
COMPUTE MANAGEMENT
Focusing on HPC In Ten Years

All procedures compiled for both host and device execution

- System calls
- Function pointers, virtual function tables
- Language runtime fully ported to device

Statically linking host and device code to same address space
Dynamic linking
Lots of creative work to be done here:

- Selecting which code sections to leave on the host
- Optimizing compute schedules (threadIdx.x/blockIdx.x mappings)
- Optimizing data layouts
- Launching asynchronous kernels
KNOWN HARD PROBLEMS

• Spread work across CPU as well as GPU (load balancing)
• Spread work across multiple GPUs
UNKNOWN UNKNOWNS
What capabilities will GPUs have in ten years

2007
• texture cache: dedicated texture cache, hardware interpolation

2017
• tensor cores: 4x4 HP matmul with 4x4 SP matadd

2027
• ?
Choosing which parallel constructs to offload?
Optimizing the mapping to GPU thread and grid indices?
Controlling placement of data in memory spaces?
Mapping to asynchronous operations?
Satisfying Lisa the Library Writer as well as Phil the Physicist?