Programming Heterogeneous X64+GPU Systems Using OpenACC

Michael Wolfe, Compiler Engineer
The Portland Group, Inc.

www.pgroup.com
What is OpenACC?

A set of directive-based extensions to C, C++ and Fortran that allow you to annotate regions of code and data for offloading from a CPU host to an attached Accelerator.
Technical Computing

- Automotive
- Aerospace
- Financial
- Medical
- Nuclear simulation
- Cosmology
- Combustion
- Environmental
- Weather, Climate


http://sitemaker.umich.edu/saiprasad/cool_cfd_simulations

http://www.research.noaa.gov/climate/t_modeling.html

http://www.cerfacs.fr/4-26780-Piston-engine.php
## Supercomputers

<table>
<thead>
<tr>
<th></th>
<th>Fujitsu K</th>
<th>IBM Sequoia</th>
<th>Cray Titan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cabinets</td>
<td>864</td>
<td>96</td>
<td>200</td>
</tr>
<tr>
<td>Nodes</td>
<td>82944</td>
<td>98304</td>
<td>18688</td>
</tr>
<tr>
<td>Cores</td>
<td>705024</td>
<td>1572864</td>
<td>299008</td>
</tr>
<tr>
<td>CPU</td>
<td>SPARC</td>
<td>PowerPC</td>
<td>AMD+Kepler</td>
</tr>
<tr>
<td></td>
<td>multicore</td>
<td>embedded</td>
<td>accelerated</td>
</tr>
<tr>
<td>Power</td>
<td>12.6MW</td>
<td>7.8MW</td>
<td>8.2MW</td>
</tr>
<tr>
<td>Top500</td>
<td>#3</td>
<td>#2</td>
<td>#1</td>
</tr>
<tr>
<td>Rmax</td>
<td>10.5PF</td>
<td>16.3PF</td>
<td>17.5PF</td>
</tr>
</tbody>
</table>

Photos: Top500.org
CPU+Accelerator
Abstract Machine Architecture

Multicore CPU

Latency Optimized Host Memory

Execution Queues

Control

PE O

PE 1

PE n-1

SIMD/SIMT

Hardware/Software Cache

Hardware/Software Cache

Hardware/Software Cache

Stream Optimized Device Memory

©2012 The Portland Group, Inc.
How to make a faster CPU

- Faster clock
- More work per clock
  - Pipelining
  - Multiscalar instruction issue, VLIW
  - Vector / SIMD instructions
  - More cores
- Fewer stalls
  - Cache memories
  - Branch prediction
  - Reservation stations, out-of-order execution
  - Multithreading
How is a GPU different?

**CPU**
- Faster clock (2.5-3.5 GHz)
- More work per clock
  - Pipelining (deep)
  - Multiscalar (3-4)
  - SIMD instructions (4-16)
  - More cores (6-12)
- Fewer stalls
  - Large cache memories
  - Branch prediction
  - Out-of-order execution
  - Multithreading (2-4)

**GPU**
- Slower clock (0.8-1.0 GHz)
- More work per clock
  - Pipelining (shallow)
  - Multiscalar (1-2)
  - SIMD instructions (16-64)
  - More cores (15-32)
- Fewer stalls
  - Small cache memories
  - Little branch prediction
  - In-order execution
  - Multithreading (15-32)
CPU+Accelerator
Abstract Machine Architecture
GPU Programming Issues

- **Performance**
  - Memory management
  - Parallelism management
  - Data access patterns

- **Portability**
  - From CPU to GPU
  - From GPU to another GPU
  - Performance across GPUs
  - Performance on future GPUs

- **Productivity**
GPU Programming Solutions

- Low-Level Languages
  - CUDA, OpenCL
- Libraries
  - MAGMA, Thrust, CULATools, …
- High-Level Directives
  - OpenACC
OpenACC Directives

```c
#pragma acc data copyin(in[0:n]) copyout(out[0:n]) \
    copy(force[0:n], vel[0:n])
{
...
}
```
OpenACC Directives

#pragma acc data copyin(in[0:n]) copyout(out[0:n]) \
copy(force[0:n], vel[0:n])
{
    #pragma acc parallel loop
    for (int i = 0; i < n; i++)
    {
        // update forces
    }
    #pragma acc parallel loop
    for (int i = 0; i < n; i++)
    {
        // update positions, velocities
    }
}
OpenACC 1.0 Features

- Single source code for CPU and GPU
- Offload loops and data with directives
- Incrementally tune data movement
- Overlap data movement with computation
- Re-use Accelerator data across kernels, even across procedure calls
- Easy to experiment with alternative loop schedules, mapping of parallelism to HW
OpenACC 2.0
Upcoming Features

- Procedure calls on the Accelerator
- Unstructured Accelerator data lifetimes
- Nested parallelism
- Atomic operations
- Better interaction with OpenMP parallelism
- and more...
Programming Heterogeneous X64+GPU Systems Using OpenACC

- Technical computing benefits from more compute, more memory bandwidth
- Cost, energy are increasingly the limiting factors
- Accelerators take advantage of parallelism, regularity
  - expose, express, exploit
  - algorithm, language, compiler + runtime + hardware
- Look for follow-on IEEE webinar later this year
  
  www.pgroup.com/openacc
  www.openacc.org
Save 50%

PGI Accelerator™ Fortran/C/C++ Workstation
Compiler Suite with OpenACC

$749 commercial/gov’t ($1,499 list)

$349 academic ($799 list)

Offer valid through Friday, May 31, 2013 for registered webinar attendees, limit one copy per attendee

30 day money back guarantee

E-mail PGI Sales at sales@pgroup.com from your registered e-mail address and reference offer code IEEEACC