**MIC-GPU:** High-Performance Computing for Medical Imaging on Programmable Graphics Hardware (GPUs)

**Closing Remarks**

Klaus Mueller and Sungsoo Ha  
Stony Brook University  
Computer Science  
Stony Brook, NY

**OpenCL**

OpenCL: Open Computing Language (based on C)

- support for heterogeneous devices (GPU, CPU, …)
- pick the device best suited for the job
- potential parallelism is key for selection
- recall Amdahl's law

**OpenCL Mindset**

Platform model:

- a host is connected to one or more OpenCL devices
- a device is divided into one or more compute units (cores)
- compute units are divided into one or more processing elements

**Execution Model**

- host programs execute on the host
- kernels execute on one or more OpenCL devices
- each instance of a kernel is called a work item
- work items are organized as work groups
- work groups and work items are defined into an *index space*
- index space is created upon kernel submission
- work items can be identified by work group and local work item IDs

→ this is all quite similar to CUDA

<table>
<thead>
<tr>
<th>CUDA Terminology</th>
<th>OpenCL Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>Index Space</td>
</tr>
<tr>
<td>Block</td>
<td>Work Group</td>
</tr>
<tr>
<td>Thread</td>
<td>Work Item</td>
</tr>
</tbody>
</table>
Global and Local Dimensions

Synchronization between work-items possible only within workgroups: barriers and memory fences

Cannot synchronize outside of a workgroup

OpenCL Memory Model

Private memory
- per work item

Local memory (16kB)
- shared per work group

Global/constant memory
- not synchronized

Host memory
- on CPU

Recommended Literature

text book reference books

programming guides available from nvidia.com

more general books on parallel programming
References


K. Mueller, F. Xu, and N. Neophytou, "Why do GPUs work so well for acceleration of CT?," SPIE Electronic Imaging '07 (Keynote, Computational Imaging V), 54980N, San Jose, January 2007.

