Heterogeneous Parallel Programming

Related Programming Models
OpenCL Device Architecture

Wen-mei Hwu – University of Illinois at Urbana-Champaign
Objective

• To Understand the OpenCL device architecture
  • Foundation to terminology used in the host code
  • Also needed to understand the memory model for kernels
OpenCL Hardware Abstraction

- OpenCL exposes CPUs, GPUs, and other Accelerators as “devices”
- Each device contains one or more “compute units”, i.e. cores, Streaming Multicprocessors, etc...
- Each compute unit contains one or more SIMD “processing elements”, (i.e. SP in CUDA)
OpenCL Device Architecture

Compute Device

Compute unit 1

Private memory 1

PE 1

Local memory 1

Private memory M

PE M

Compute unit N

Private memory 1

PE 1

Local memory N

Private memory M

PE M

Global/Constant Memory Data Cache

Global Memory

Constant Memory

Compute Device Memory
### OpenCL Device Memory Types

<table>
<thead>
<tr>
<th>Memory Type</th>
<th>Host access</th>
<th>Device access</th>
<th>CUDA Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>global memory</td>
<td>Dynamic allocation; Read/write access</td>
<td>No allocation; Read/write access by all work items in all work groups, large and slow but may be cached in some devices.</td>
<td>global memory</td>
</tr>
<tr>
<td>constant memory</td>
<td>Dynamic allocation; read/write access</td>
<td>Static allocation; read-only access by all work items.</td>
<td>constant memory</td>
</tr>
<tr>
<td>local memory</td>
<td>Dynamic allocation; no access</td>
<td>Static allocation; shared read-write access by all work items in a work group.</td>
<td>shared memory</td>
</tr>
<tr>
<td>private memory</td>
<td>No allocation; no access</td>
<td>Static allocation; Read/write access by a single work item.</td>
<td>registers and local memory</td>
</tr>
</tbody>
</table>
OpenCL Context

- Contains one or more devices
- OpenCL device memory objects are associated with a context, not a specific device
Heterogeneous Parallel Programming

TO LEARN MORE, READ SECTION 14.3