Starting to Learn to Program GPUs with OpenMP

• A GPU Common Core
• The hierarchy
• Learning OpenMP at Supercomputing’23
### The GPU Common Core

<table>
<thead>
<tr>
<th>OpenMP pragma, clause, or environment var.</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#pragma omp target</td>
<td>Offload execution to a target device</td>
</tr>
<tr>
<td>#pragma omp loop</td>
<td>Run the following loop in parallel</td>
</tr>
<tr>
<td>reduction(op: list)</td>
<td>Reduction using <code>op</code> for variables in <code>list</code></td>
</tr>
<tr>
<td>collapse(n)</td>
<td>Combine <code>n</code> nested loops into one logical loop</td>
</tr>
<tr>
<td>map([to</td>
<td>from</td>
</tr>
<tr>
<td>#pragma omp target data</td>
<td>Manage data on a device for a structured block</td>
</tr>
<tr>
<td>#pragma omp target update to(list)</td>
<td>Update data <code>to</code> or <code>from</code> a device</td>
</tr>
<tr>
<td>#pragma omp target update from(list)</td>
<td>Update data <code>to</code> or <code>from</code> a device</td>
</tr>
<tr>
<td>#pragma omp target enter data</td>
<td>Move data into/from a device data environment</td>
</tr>
<tr>
<td>#pragma omp target exit data</td>
<td>Move data into/from a device data environment</td>
</tr>
<tr>
<td>#pragma omp target enter data</td>
<td>Move data into/from a device data environment</td>
</tr>
<tr>
<td>#pragma omp target exit data</td>
<td>Move data into/from a device data environment</td>
</tr>
<tr>
<td>OMP_TARGET_OFFLOAD=mandatory</td>
<td>Force target region to execute on a device</td>
</tr>
</tbody>
</table>

Table from Deakin & Mattson, Programming Your GPU with OpenMP, MIT Press
**#pragma omp target**

- Directive has two roles:
  - transfer execution to the device
  - transfer data to/from the device

- Implicit data transfers:
  - firstprivate: scalars
  - copied to and from (map(tofrom)): stack arrays, complete structs
  - zero-length arrays: pointers
    - (pointer itself is firstprivate if in a map clause)

---

Figure from Deakin & Mattson, Programming Your GPU with OpenMP, MIT Press
```c
#pragma omp loop
#pragma omp target
#pragma omp loop
for (int i = 0; i < N; ++i){
    // body
}
```

- Loop directive says loop iterations are concurrent
- Places restrictions on loop body
  - No API calls, barriers, etc.
- Loop has binding rules, but...
- ...compiler can parallelise across whole device by “as if” rule
The map clause

float *A, *B, *C;
// allocate and init ...
#pragma omp target \
    map(to: A[0:N], B[0:N]) \
    map(from: C[0:N])
#pragma omp loop
for (int i = 0; i < N; ++i){
    C[i] = A[i] + B[i];
}

• Implicit rules are always in operation
• Use map() clause to map data allocated on the heap
• Use OpenMP array syntax
  – [start:length]
• Pointers are attached
  – pointer value is firstprivate, cannot be changed
Persistent data

```c
#pragma omp target data map(...) 
{
    #pragma omp target
    { // code }
    #pragma omp target
    { // code }
}

#pragma omp target enter data
#pragma omp target update
#pragma omp target exit data
```

- Keeping data resident on the device is a key for high performance
- Map data into device data environment with target data directives
- Cannot use the mapped arrays on the host without synchronization
  - Use target update directive
Hierarchical Parallelism

- GPU hardware is hierarchical
- Can program at this level with OpenMP:
  - #pragma omp teams
  - #pragma omp distribute
  - #pragma omp parallel for
  - (#pragma omp simd)
- Or use loop bind() clauses

Figures from Deakin & Mattson, Programming Your GPU with OpenMP, MIT Press
Using OpenMP - The Next Step

- Available from MIT Press
  – ISBN: 9780262534789
- Covers OpenMP 4.5
- Chapters on tasking, NUMA, GPUs, and SIMD
Programming Your GPU with OpenMP

• Available now from MIT Press
  – ISBN: 9780262547536
• Up to date with OpenMP 5.2
• https://ompgpu.com
Learn OpenMP at Supercomputing

- The OpenMP Common Core: A Hands-on Introduction
  - Sunday, 12 November, 8:30am – 5pm MST
  - *Tim Mattson, Yun (Helen) He, Alice Koniges, David Eder*

- Mastering Tasking with OpenMP
  - Monday, 13 November, 8:30am – Noon MST
  - *Christian Terboven, Michael Klemm, Xavier Teruel, Bronis R. de Supinski*

- Advanced OpenMP: Host Performance and 5.2 Features
  - Monday, 13 November, 1:30pm – 5pm MST
  - *Christian Terboven, Michael Klemm, Ruud van der Pas, Bronis R. de Supinski*

- Programming Your GPU with OpenMP: A hands-on Introduction
  - Monday, 13 November, 8:30am – 5pm MST
  - *Tom Deakin, Tim Mattson*
OpenMP SC’23 Booth Talk Series

openmp.org
OpenMP API specs, forum, reference guides, and more

link.openmp.org/sc23
OpenMP SC’23 booth talk videos and presentations