Enabling Legacy Applications on Heterogeneous Platforms
Michela Becchi, Hari Cadambi and Srimat Chakradhar
NEC Laboratories America, Princeton, NJ, USA.

Context
Legacy applications that are retargeted to heterogeneous (CPU + multiple accelerators) systems using library interposing

Goals
- Schedule kernels to multiple accelerators considering both computation and data
- Improve performance by reducing data movement.

Strategy
- Profile and estimate each kernel’s performance on each accelerator.
- Schedule next computation taking into consideration (i) estimated performance of the accelerator and (ii) cost to move data to the accelerator.

System

Enforcing coherence using OS
- Associate each process with a set of data blocks
- Similar to Linux associating each process with a set of memory regions
- Each data block stores accelerator mapped addresses, sizes of the memory regions, location information and whether data is synchronized

### Results

**Sort and Reduce**

<table>
<thead>
<tr>
<th>Number of Elements</th>
<th>32 queries</th>
<th>64 queries</th>
<th>96 queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.5</td>
<td>2.0</td>
<td>2.5</td>
</tr>
<tr>
<td>1</td>
<td>1.0</td>
<td>1.5</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>0.5</td>
<td>1.0</td>
<td>1.5</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
<td>0.5</td>
<td>1.0</td>
</tr>
</tbody>
</table>

**Semantic Search**

- 1.6M documents
  - Sort and Reduce
  - data transfer
  - topk_rank
  - sgemm

Example

- `matmul(in A, in B, out C, sizes...)`
  - Model says GPU performance is better. So transfer A, B, C and schedule on GPU. Leave C on GPU.
- `matmul(in C, in D, out E, sizes...)`
  - Model says CPU performance is better, but C too big to transfer back, so schedule on GPU. Leave C, E on GPU. CPU accesses C, E which are stale since latest copies are on GPU. Must force synchronization now for coherence.
- `func(C, E)`
  - CPU accesses C, E which are stale since latest copies are on GPU. Must force synchronization now for coherence.