Global Climate Modeling

- Understand future local sea level changes
- Needs high-resolution simulations
- Combine two approaches:
  - Distributed computing (multiple resources)
    - Ibis couples models for land, ice, ocean, atmosphere
  - GPUs

COMMIT/
Enlighten Your Research Global award

STAMPEDE (USA)

KRAKEN (USA)

CARTESIUS (NLD)

EMERALD (UK)

SUPERMUC (GER)

#7

10G

10G

10G

10G

#10
GPU Computing

- Offload expensive kernels for Parallel Ocean Program (POP) from CPU to GPU
  - Many different kernels, fairly easy to port to GPUs
  - Execution time becomes virtually 0
- New bottleneck: moving data between CPU & GPU
Different methods for CPU-GPU communication

- Memory copies (explicit)
  - No overlap with GPU computation
- Device-mapped host memory (implicit)
  - Allows fine-grained overlap between computation and communication in either direction
- CUDA Streams or OpenCL command-queues
  - Allows overlap between computation and communication in different streams
- Any combination of the above
Problem

- Problem:
  - Which method will be most efficient for a given GPU kernel? Implementing all can be a large effort

- Solution:
  - Create a performance model that identifies the best implementation:
    - What implementation strategy for overlapping computation and communication is best for my program?

Ben van Werkhoven, Jason Maassen, Frank Seinstra & Henri Bal: Performance models for CPU-GPU data transfers, CCGrid2014 (nominated for best-paper-award)
Example result

Measured and estimated performance on GTX 680 (PCIe 2.0)

- **Measured**
  - explicit: red
  - implicit: green
  - streams: blue
  - hybrid: purple

- **Model**
  - buoydiff-observed: red
  - buoydiff-estimated: blue

Bar chart showing time (ms) for different configurations.
Different GPUs (state kernel)
Different GPUs (buoydiff)
**Comes with spreadsheet**

### Performance models for CPU-GPU data transfers

<table>
<thead>
<tr>
<th><strong>1) GPU properties</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of copy engines</td>
<td><strong>No</strong></td>
</tr>
<tr>
<td><strong>2) Kernel properties</strong></td>
<td></td>
</tr>
<tr>
<td>Host to Device transfers (bytes)</td>
<td>538870912</td>
</tr>
<tr>
<td>Kernel Execution Time (ms)</td>
<td>20</td>
</tr>
<tr>
<td>Device to Host transfers (bytes)</td>
<td>0</td>
</tr>
<tr>
<td>Average # loads of input elements per thread</td>
<td>1</td>
</tr>
<tr>
<td><strong>2b) Streams properties (optional, leave blank if unused)</strong></td>
<td></td>
</tr>
<tr>
<td>Largest per stream HtoD transfer (bytes)</td>
<td></td>
</tr>
<tr>
<td>Largest per stream DtoH transfer (bytes)</td>
<td></td>
</tr>
<tr>
<td>Number of streams</td>
<td></td>
</tr>
<tr>
<td><strong>3) Interconnect parameters</strong></td>
<td></td>
</tr>
<tr>
<td>Load preset</td>
<td>GTX Titan</td>
</tr>
<tr>
<td>(do not edit below this line)</td>
<td></td>
</tr>
</tbody>
</table>

---

#### Estimated kernel performance

- Explicit
- Implicit
- Streams
- Hybrid

**Computation / Communication balance**

- 42%
- 18%
- 40%

**Theoretical bounds**

- Varying the number of streams

---

For more information about the performance models and parameters please see:

- Performance models for CPU-GPU data transfers
- B. van Werkhoven, J. Maassen, F.J. Seemstra, and H.E. Bal