ISx: Integer Sort Mini-Application for the Exascale Era

Jacob Hemstad
Ulf Hanebutte
September 17th, 2015
NAS Integer Sort

- Released 1991 – C & Fortran
- **THE** parallel integer sort benchmark
- OpenSHMEM port 2012 - UofH
NAS Integer Sort

- **Drawbacks**
  - Strong scaling only
  - Upper limit of $2^{31}$ total keys
  - Hardcoded 1024 buckets
  - Non-uniform key distribution
    - Realistic, but load balancing is ineffective
  - Complicated code

**NAS Load Balancing**

- Number of Keys (millions) vs. PE
- Graph showing decrease in number of keys with increasing PE
ISx: A Modern Integer Sort Benchmark

- OpenSHMEM
- Modular, ‘strict’ software design
- Uniform key distribution (for now)
  - Uniform sized buckets
  - Load balancing is orthogonal to current scope
- No limit on number of keys or PEs
  - # buckets = # PEs

- Verification
  - All keys within bucket boundaries
  - Same # of keys before and after sort

- Two scaling options
  - Strong
    - N = constant
  - Weak
    - N/P = constant
## Code Complexity

<table>
<thead>
<tr>
<th></th>
<th>NAS</th>
<th>ISx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lines of Code</td>
<td>679</td>
<td>505</td>
</tr>
<tr>
<td># of Functions</td>
<td>6</td>
<td>22</td>
</tr>
<tr>
<td>Avg. LOC per Function</td>
<td>82</td>
<td>22</td>
</tr>
<tr>
<td>Avg. Cyclomatic Complexity*</td>
<td>16.5</td>
<td>4</td>
</tr>
</tbody>
</table>

*https://en.wikipedia.org/wiki/Cyclomatic_complexity*