2009 IBM HPC Challenge Class II Submission

George Almási  Ganesh Bikshandi  Călin Cașcaval
David Cunningham  Gábor Dózsa  Montse Farreras
David Grove  Sreedhar Kodali  Nathaniel Nystrom
Igor Peshansky  Vijay Saraswat  Sayantan Sur
Olivier Tardieu  Ettore Tiotto

2009 IBM HPCC  12/07/09
Our submission at a glance

- **Two programming languages**
  - X10
  - UPC

- **Three platforms**
  - Power 5+ cluster (Poughkeepsie Benchmark Center)
  - Blue Gene/P (instead of Blue Gene/L)
  - BSC MareNostrum

- **One common distributed runtime**
HPC programming models research at IBM

- xlUPC
  - UPC moving towards standardization
  - PERCS, BW deliverable
  - Power architectures

- xlCAF
  - CAF in Fortran2008 standard
  - Prioritized subsets in future Fortran releases

- X10
  - Open source
    - Eclipse Public License
  - X10 2.0 released November 6, 2009
    - Java or C++ back-end
    - Runs on almost any architecture

Common runtime support for all three efforts

http://www.alphaworks.ibm.com/tech/upccompiler
http://x10-lang.org
HPCC Benchmarks

- **X10:**
  - Benchmarks rewritten for X10 2.0
    - LU, FT: new scalable version
      - Use APGAS collectives
        > Broadcast, Reduction, Alltoall
    - RA, Stream:
      - Reduced overheads

- **UPC:**
  - Benchmarks *almost* unchanged from 2008
    - FFT
      - Local scatter + alltoall instead of “memput”
    - HPL
      - Reduced loop overhead (compiler opt);
      - Better optimized collectives

**HPCC submission completed even though it overlapped with PERCS milestone**

1 PW to compile, run, organize
Performance results: Power5+ cluster

<table>
<thead>
<tr>
<th>X10</th>
<th>LU</th>
<th>RA</th>
<th>Stream</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>GFlop/s</td>
<td>MUP/s</td>
<td>GBytes/s</td>
<td>GFlops/s</td>
</tr>
<tr>
<td>4</td>
<td>354</td>
<td>6.34</td>
<td>325.7</td>
<td>23.67</td>
</tr>
<tr>
<td>8</td>
<td>666</td>
<td>12.31</td>
<td>650.5</td>
<td>40.62</td>
</tr>
<tr>
<td>16</td>
<td>1268</td>
<td>23.02</td>
<td>1287.8</td>
<td>65.92</td>
</tr>
<tr>
<td>32</td>
<td>43.1</td>
<td>2601.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UPC</th>
<th>LU</th>
<th>RA</th>
<th>Stream</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>GFlop/s</td>
<td>MUP/s</td>
<td>GBytes/s</td>
<td>GFlops/s</td>
</tr>
<tr>
<td>4</td>
<td>379</td>
<td>5.5</td>
<td>140</td>
<td>7.9</td>
</tr>
<tr>
<td>8</td>
<td>747</td>
<td>10.8</td>
<td>256</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>1442</td>
<td>21.5</td>
<td>523</td>
<td>26.3</td>
</tr>
<tr>
<td>32</td>
<td>2333</td>
<td>43.3</td>
<td>1224</td>
<td>39.8</td>
</tr>
</tbody>
</table>

IBM Poughkeepsie Benchmark Center

32 Power5+ nodes
16 SMT 2x processors/node
64 GB/node; 1.9 GHz
HPS switch, 2 GBytes/s/link
Performance results – Blue Gene/P

<table>
<thead>
<tr>
<th>X10</th>
<th>LU</th>
<th>RA</th>
<th>Stream</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>GFlop/s</td>
<td>GUP/s</td>
<td>GBytes/s</td>
<td>GFlops/s</td>
</tr>
<tr>
<td>32</td>
<td>186</td>
<td>0.042</td>
<td>141</td>
<td>9.6</td>
</tr>
<tr>
<td>128</td>
<td>713</td>
<td>0.16</td>
<td>564</td>
<td>13.9</td>
</tr>
<tr>
<td>1024</td>
<td>5874</td>
<td>1.05</td>
<td>4516</td>
<td></td>
</tr>
<tr>
<td>2048</td>
<td>1.42</td>
<td>9032</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>UPC</th>
<th>LU</th>
<th>RA</th>
<th>Stream</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>GFlop/s</td>
<td>GUP/s</td>
<td>GBytes/s</td>
<td>GFlops/s</td>
</tr>
<tr>
<td>32</td>
<td>242</td>
<td>0.04</td>
<td>168</td>
<td>8.28</td>
</tr>
<tr>
<td>128</td>
<td>967</td>
<td>0.16</td>
<td>672</td>
<td>28</td>
</tr>
<tr>
<td>1024</td>
<td>7744</td>
<td>1.27</td>
<td>5376</td>
<td>246</td>
</tr>
<tr>
<td>2048</td>
<td>15538</td>
<td>2.54</td>
<td>492</td>
<td></td>
</tr>
<tr>
<td>4096</td>
<td>28062</td>
<td>5.04</td>
<td>519</td>
<td></td>
</tr>
</tbody>
</table>

IBM TJ Watson Res. Ctr. WatsonShaheen

4 racks Blue Gene/P
1024 nodes/rack
4 CPUs/node; 850 MHz
4 Gbytes/node RAM
16 x 16 x 16 torus

HPL perf. comparison

RA perf. comparison
### Performance results – MareNostrum

<table>
<thead>
<tr>
<th>UPC</th>
<th>LU</th>
<th>Stream</th>
<th>FFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>nodes</td>
<td>GFlop/s</td>
<td>GBytes/s</td>
<td>GFlops/s</td>
</tr>
<tr>
<td>2x2</td>
<td>20</td>
<td>12.3</td>
<td>0.37</td>
</tr>
<tr>
<td>4x2</td>
<td>38</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>8x2</td>
<td>76</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>16x2</td>
<td>141</td>
<td>2.73</td>
<td></td>
</tr>
<tr>
<td>32x2</td>
<td>276</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>64x2</td>
<td>541</td>
<td>391.05</td>
<td>11</td>
</tr>
<tr>
<td>128x2</td>
<td>1003</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>256x2</td>
<td>1885</td>
<td>1558</td>
<td>40</td>
</tr>
<tr>
<td>256x4</td>
<td>2709</td>
<td>53</td>
<td></td>
</tr>
</tbody>
</table>

**BSC MareNostrum**

- CPUs: 2x2x2560 PPC 970MP
- 2.3 GHz, 8 GB/node
- Network: Myrinet 2Gb/s crossbars
- 10 cabinets 256x256
- 2 “spines” 1280x1280

GUPS runs prevented by network failures
Discussion

Platforms

- Power5+:
  - Forgiving machine
  - Performance, scalability is easy
  - Firmware limits RA performance
- BG/P:
  - Hard memory limitations
  - X10 scaling achievable
- MareNostrum:
  - Network issues

Benchmarks

- LU:
  - Global view (UPC)
  - Explicit blocking; SPMD
  - APGAS collectives
- FT:
  - Scatter/transpose algorithm
  - Alltoall collective
- RA:
  - Network performance
  - Low runtime overhead
- Stream:
  - Rely on back-end compiler

Productivity in HPC:
Programs are easy to write;
**High performance** programs are easy to write.
Our thanks to:

- IBM Poughkeepsie Benchmark Center: S. Selzo
- NCSA BluePrint cluster: M. Showerman, W. Gropp
- IBM Research/Watson Shaheen: F. Mintzer, D. Singer, A. Raishubsky, B. Fitch
- BSC: David Vicente
- Christian Bell (Myricom)
Backup
Performance factors

- **Runtime:**
  - Low overhead
  - Collective communication
  - Good language support (finish, async, shared arrays)

- **Compiler optimization is crucial**
  - UPC:
    - Locality inference
    - Comm. Aggregation
  - X10:
    - Allocation optimization
    - async/finish optimization
X10: Innovation, Productivity, Scalability

Fine grained concurrency
- async S

Atomicity
- atomic S
- when (c) S

Place-shifting operations
- at (P) S

Ordering
- finish S
- clock

Global data-structures
- points, regions, distributions, arrays

Two basic ideas: Places and Asynchrony
APGAS: one library to run them all

- Support for UPC and CAF
  - shared arrays; pointers-to-shared; locks; optimized collectives
- Support for X10
  - Asynchs & activities; remote references
- Multiplatform
  - Power, BG, Intel, Sun etc.
  - LAPI (IB, HPS), DCMF (BG), MX (Myrinet), TCP/IP sockets
- Interoperable
  - MPI
<table>
<thead>
<tr>
<th># Nodes</th>
<th>Torus</th>
<th>Bisection (links)</th>
<th>BW (GB/s)</th>
<th>GUPS limit</th>
<th>FFT limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>4x4x2</td>
<td>32</td>
<td>13.6</td>
<td>0.32</td>
<td>39</td>
</tr>
<tr>
<td>128</td>
<td>4x8x4</td>
<td>128</td>
<td>55</td>
<td>1.30</td>
<td>176</td>
</tr>
<tr>
<td>1024</td>
<td>8x8x16</td>
<td>256</td>
<td>109</td>
<td>2.6</td>
<td>870</td>
</tr>
<tr>
<td>2048</td>
<td>8x16x16</td>
<td>512</td>
<td>217</td>
<td>5.2</td>
<td>1741</td>
</tr>
<tr>
<td>4096</td>
<td>16x16x16</td>
<td>1024</td>
<td>434</td>
<td>10.3</td>
<td>1562</td>
</tr>
</tbody>
</table>

Torus bisection = smallest diameter x 2 (torus) x 2 (half traffic)
Bisection Bandwidth = Bisection x 0.42 GB/s/link
GUPS limit = Bisection bandwidth / 42 bytes/packet
FFT Gflops = flops * BW / Bytes
FFT Gflops = 5* log(N) * N * N * Bandwidth / 3 * N * N * sizeof(cplx)