A Comparison of Task Mapping Strategies on Two Generations of Cray Systems

February 18, 2014
SIAM Conference on Parallel Processing

Kevin Pedretti
CS R&D Technical Staff
Scalable System Software
Sandia National Laboratories

Torsten Hoefler
Assistant Professor
Scalable Parallel Computing Lab
ETH Zürich
Outline

- Motivation
- Modeling Cray Gemini Network as a Graph (XE6 / XK7)
  - Topology
  - Static Routing Info
- MiniGhost Task Mapping Results
- Cray Aries XC30 Preview
- Conclusions
Why Task Mapping?

- **Increase performance**
  - By reducing the distance a message travels, its latency is reduced and it has less chance of competing with other messages for bandwidth
  - Minimize volume of communication => less network congestion
  - Net bandwidth / compute ratio getting much worse, scarce resource

- **Reduce power (i.e., the performance bottleneck)**
  - Data movement is energy intensive... move data as little as possible
  - Being oblivious to task mapping drives over-engineering of network, driving up both network power and system cost

- Put pressure on system software developers (like me) to implement task mapping interfaces (e.g., MPI graph comms)

*Task Mapping is Important both Intra-Node and Inter-Node*
## Scalable Networks Are Sparse

<table>
<thead>
<tr>
<th>Period</th>
<th>System Name</th>
<th>Architecture</th>
<th>Dimensions</th>
<th>Nodes</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997 – 2006</td>
<td>SNL ASCI Red</td>
<td>3-D Mesh</td>
<td>38 x 32 x 2</td>
<td>4510</td>
<td>3.15 TFLOPS/s</td>
</tr>
<tr>
<td></td>
<td>Intel Custom Network</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004 - 2012</td>
<td>SNL Red Storm</td>
<td>3-D Mesh</td>
<td>27 x 20 x 24</td>
<td>12960</td>
<td>284 TFLOP/s</td>
</tr>
<tr>
<td></td>
<td>Cray XT3 SeaStar</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011 –</td>
<td>ACES Cielo</td>
<td>3-D Torus</td>
<td>16 x 12 x 24</td>
<td>8944</td>
<td>1374 TFLOP/s</td>
</tr>
<tr>
<td></td>
<td>Cray XE6 Gemini</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013 –</td>
<td>NERSC Edison</td>
<td>Dragonfly</td>
<td>3-Levels: 16, 6, 14</td>
<td>5192</td>
<td>2390 TFLOP/s</td>
</tr>
<tr>
<td></td>
<td>Cray XC30 Aries</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Total BW / Injection BW Ratios

<table>
<thead>
<tr>
<th>Period</th>
<th>System Name</th>
<th>Node Injection (GB/s)</th>
<th>Network (TB/s)</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997 – 2006</td>
<td>SNL ASCI Red</td>
<td>1443</td>
<td>4752</td>
<td>3.3</td>
</tr>
<tr>
<td>2004 - 2012</td>
<td>SNL Red Storm</td>
<td>22</td>
<td>357</td>
<td>16.2</td>
</tr>
<tr>
<td>2011 –</td>
<td>ACES Cielo</td>
<td>55</td>
<td>281</td>
<td>5.1</td>
</tr>
<tr>
<td>2013 –</td>
<td>NERSC Edison</td>
<td>48</td>
<td>156 – 204</td>
<td>3.3 – 4.25</td>
</tr>
</tbody>
</table>

---

Total Node Injection: 1443 GB/s  
Total Network (all links): 4752 GB/s  
Ratio: 3.3
Example Case of “Bad” Task Mapping

- MiniGhost is a proxy application, represents CTH full application
- Explicit time-stepping, synchronous communication, 27-point stencil across 3-D grid
- Dark Red Curve: Original configuration scaled poorly after 16K cores (1024 nodes, 512 Geminis)
- Light Red Curve: Reorder MPI rank to node mapping to reduce off-node communication
  
  Original: 1x1x16 ranks/node
  Reorder: 2x2x4 ranks/node

Interconnect is a 3-D torus. Application talks to nearest 3-D neighbors. Should be match made in heaven, So what’s going on?
Outline

- Motivation
- Modeling Cray Gemini Network as a Graph (XE6 / XK7)
  - Topology
  - Static Routing Info
- MiniGhost Task Mapping Results
- Cray Aries XC30 Preview
- Conclusions
Wanted to Try Libtopomap on Cray

- Task mapping library created by Torsten Hoefler
  - Graph based, both app and system represented as a graph
  - Several strategies to map app graph to system graph
    - Simple greedy, greedy considering routes, recursive bisection, graph similarity (Reverse Cuthill McKee), SCOTCH adapter, multicore partitioning, simulated annealing, ..

- Had to generate two input files for Libtopomap
  - topomap.txt
    - Vertices are hosts and routers, edges are network links
    - Directed graph, edge weights represent link speed
  - routes.txt (Cray specific extension)
    - X,Y,Z coordinate of each node
    - Static route from each source host to each destination host
  - Run some scripts to generate once per system, use many times

*Hoefler and Snir: Generic Topology Mapping Strategies for Large-scale Parallel Architectures, ICS’11*
Application Provides Communication Graph for an Existing Communicator

New Communicator with Ranks Permuted (Application must move data around accordingly)

Grey = Internal to MPI Library
System Topology Model

P0 P1 P2 P3

N0 N1 N2 N3

Task Mapping Algorithm (libtopomap)
Cray Gemini Interconnect

- Two nodes (hosts) per Gemini chip
- Gemini chip consists of:
  - Two network interfaces
  - 48 port router (48 “tiles”)  
- Gemini router ports organized into groups to form seven logical links
  - X+, X-, Y+, Y-, Z+, Z-, Host
  - XYZ links connected to neighbor Gemini chips to form 3-D torus
- Large set of performance counters
  - NIC and router counters
  - Cray Documentation (S-0025-10): *Using the Cray Gemini Hardware Counters*
Calculating Edge Weights

- Get map of each Gemini’s 48 tiles from Cray database
- Link speeds are heterogeneous (!)

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mezzanine</td>
<td>2.34 GB/s</td>
</tr>
<tr>
<td>Backplane</td>
<td>1.88 GB/s</td>
</tr>
<tr>
<td>Cable</td>
<td>1.17 GB/s</td>
</tr>
<tr>
<td>Host</td>
<td>1.33 GB/s (est.)</td>
</tr>
</tbody>
</table>

### Unidirectional Bandwidths

**X Links, all:**

\[ 8 \times 1.17 = 9.4 \text{ GB/s} \]

**Y Links, alternate every other:**

\[ 4 \times 2.34 = 9.4 \text{ GB/s (mezz)} \]
\[ 4 \times 1.17 = 4.7 \text{ GB/s} \]

**Z Links, every eighth slower:**

\[ 8 \times 1.88 = 15 \text{ GB/s (backpl)} \]
\[ 8 \times 1.17 = 9.4 \text{ GB/s} \]
Cray Gemini Physical Packaging

1. Board = 1 x 2 x 1
   - 4 Nodes Per Board
   - 2 Gemini’s per Board

2. Cage = 1 x 2 x 8
   - 8 Boards per Cage

3. Cabinet = 1 x 2 x 24
   - 3 Cages Per Cabinet

- LANL / SNL Cielo XE6
  - 96 Cabinets (16 x 6 grid)
  - 16 x 12 x 24 Torus
  - 4608 Gemini chips
  - 9216 Nodes (8944 Compute)

- NCSA Blue Waters XE/XK
  - 288 Cabinets (24 x 12 grid)
  - 24 x 24 x 24 Torus
  - 13824 Gemini chips
  - 27648 Nodes (26864 Comp.)

- ORNL Titan XK7
  - 200 Cabinets (25 x 8 grid)
  - 25 x 16 x 24 Torus
  - 9600 Gemini chips
  - 19200 Nodes (18688 Comp.)
Determining Static Routing Scheme

- Performed experiments to verify empirical counters matched routes output by "rtr --logical-routes" command

- Static routing
  - All packets from a given src to dst always travels the same path
  - The path from (src to dst) not the same as (dst to src) in general
    - Request and response packets follow different paths

- All routes completely traverse the X dimension, then completely traverse Y dimension, then Z last
  - More flexible routing if there are link failures, didn’t verify
  - Should consider PUT ACK + GET REPLY backflows in system models
Cielo Cray XE6 topomap.txt

num: 13824w
# Mapping of each vertex to hostname or gemini name
0 nid00000 # host 0
1 nid00001 # host 1
2 nid00002 # host 2
3 nid00004 # host 4
[...]
9216 c0-0c0s0g0 # gemini 0
9217 c0-0c0s1g0 # gemini 1
[...
# Start of adjacency lists, one per vertex
0 9216(104) # host 0 to gemini 0 link
1 9216(104) # 2nd host gemini 0
2 9217(104) # host 2 to gemini 1 link
3 9217(104) # 2nd host gemini 1
[...]
# Start of gemini adjacency lists, each has 2 host edges and 6 net edges
9216 0(104) 1(104) 9217(150) 9239(93) 9263(93) 9503(46) 9791(93) 13823(93)
9217 2(104) 3(104) 9216(150) 9218(150) 9262(93) 9502(46) 9790(93) 13822(93)
[...]

- 9216 Nodes, 4608 Geminis, 13824 Vertices, 46080 edges (27648 net edges)
- Net edge Hist.: 4608 4.6 GB/s (Y), 14976 9.3 GB/s (XYZ), 8064 15 GB/s (Z)
- 746 KB file
Outline

- Motivation
- Modeling Cray Gemini Network as a Graph (XE6 / XK7)
  - Topology
  - Static Routing Info
- MiniGhost Task Mapping Results
- Cray Aries XC30 Preview
- Conclusions
MiniGhost Performance

- MiniGhost configuration
  - Bulk synchronous mode
  - 27-point stencil 3-D grid
  - Weak scaling mode
  - Avg. of 5 production runs, error bars stddev

- Still analyzing Libtopomap results, debugging ongoing

- Observations
  - Reordering for multicore important, still upticking (“Group”)
    - Minimize surface area by putting 2x2x4 subprob per node vs. 1x1x16
  - Leveraging geometric information pays off in this case (Mehmet’s talk)
    - But, not all applications will have geometric information
  - Libtopomap’s recursive bisection strategy is its best in this case, similar to reordering for multicore (LT uses Parmetis internally to do multicore ordering)
  - Greedy with routing is slightly better than without
    - Likely something wrong with Greedy strategy on Cray, still investigating
Correlations: Modeled vs. Measured

- Used Cray Gemini’s perf. counters to measure network congestion empirically
  - Stall counter incremented when packet can not move towards destination
  - Maximum stall count among all links (X+/-, Y+/-, Z+/-, Host)
- Max stall metric found to have best correlation to max comm time, modeled (calculated) max congestion slightly worse
  - Interference from other jobs
  - All messages are not transferred simultaneously
  - Heterogeneous link speeds in the network, for which model does not consider
Gemini Router Performance Counters

Average Bytes Injected, All Links

Average Stall Count, All Links

Link with Max Bytes Injected

Link with Max Stall Count

Outline

- Motivation
- Modeling Cray Gemini Network as a Graph (XE6 / XK7)
  - Topology
  - Static Routing Info
- MiniGhost Task Mapping Results
- Cray Aries XC30 Preview
- Conclusions
Cray Aries Interconnect

Cray Aries Blade

1. Chassis
- 16 Blades Per Chassis
- 16 Aries, 64 Nodes
- All-to-all Electrical Backplane

2. Group
- 6 Chassis Per Group
- 96 Aries, 384 Nodes
- Electrical Cables, 2-D All-to-All

3. Global
- Up to 241 Groups
- Up to 23136 Aries, 92544 Nodes
- Optical Cables, All-to-All between Groups

Gemini: 2 nodes, 62.9 GB/s routing bw
Aries 4 nodes, 204.5 GB/s routing bw

Aries has advanced adaptive routing
Ran with 16 cores per node

8192 procs = 512 nodes = 128 Aries chips

For Geom, treat Dragonfly as a 14 x 6 x 16 torus
- 14 groups, each group 6 by 16 2-D all-to-all

Geom doesn’t improve on multicore grouping, random bad
Conclusions

- Task mapping is important
- Devised method for building graph representation of Cray Gemini-based systems
  - Accurate edge weights
  - Exact routing information
- Demonstrated benefit for MiniGhost
  - Simple process grouping for multicore has big payoff
  - 3-D mesh app on 3-D torus network should be a good match
  - Future work to examine irregular applications
- Cray Aries / XC30 may be less sensitive to task mapping
Acknowledgements

- Bob Alverson (Cray)
- Richard Barrett (SNL)
- Jim Brandt (SNL)
- Karen Devine (SNL)
- Ann Gentile (SNL)
- Larry Kaplan (Cray)
- Vitus Leung (SNL)
- Stephen Olivier (SNL)
- Courtenay Vaughan (SNL)
- Sivasankaran Rajamanickam (SNL)