Using the Cray Gemini Performance Counters

Cray User Group Meeting
May 8, 2013

Kevin Pedretti, Courtenay Vaughan, Richard Barrett, Karen Devine, Scott Hemmert

Sandia National Laboratories
Outline

- Motivation and background
- How to access
- What they measure
- Usage example: MiniGhost rank remapping
- Conclusion
Motivation

- We had an application that was scaling well to 16K processes, then poorly afterwards (weak scaling)
- We suspected network congestion/contention was becoming an issue and wanted to quantify it empirically
- We had heard the Gemini had a nice set of performance counters that could do this

It turned out to be quite a bit of work to access the counters, seemed like a good topic to discuss at CUG
Cray Gemini

- Two nodes (hosts) per Gemini chip
- Gemini chip consists of:
  - Two network interfaces
  - 48 port (tile) router, logically organized into 7 network links
- Routers connected to form 3-D torus
  - X links between cabinets in a row
  - Y links between rows of cabinets
  - Z links within a cabinet
- Large set of performance counters
  - Cray Documentation (S-0025-10): Using the Cray Gemini Hardware Counters
  - This talk focuses on the router tile performance counters
## Available Tile Counters

- Each tile has six fixed counters:
  - 0: VC0_PHIT_CNT
    - Request VC phits
  - 1: VC1_PHIT_CNT
    - Response VC phits
  - 2: VC0_PKT_CNT
    - Request VC packets
  - 3: VC1_PKT_CNT
    - Response VC packets
  - 4: INQ_STALLS
    - Request VC input stalls
  - 5: CREDIT_STALLS
    - Request VC output stalls

- What is a phit? => 3 bytes
- What is a packet? => 8 to 32 phits (24 to 96 bytes)
- Input stalls? => Time waiting to get to output tile
- Output stalls? => Time waiting to get to next Gemini
Questions?

- **Basic**
  - How can we access the tile counters from an MPI program?
  - How do we turn the individual tile counters into link counters?
  - How do we calculate the capacity of each link?

- **Operational:**
  - What exactly are the packet/phit counters measuring?
  - Do the counters work as expected for PUT/GET transactions?
  - Are measurements repeatable?
  - How is the system routed?
  - Do the stall counters correlate with network congestion?
Outline

- Motivation and background

- How to access
  - What they measure
  - Usage example: MiniGhost rank remapping

- Conclusion
Directly Accessing Gemini Counters

```
gpcd_context_t *ctx;
gpcd_mmr_desc_t *desc;
gpcd_mmr_list_t *p;
int i, j, k;
char name[128];

// Create a counter group, all tile counters
ctx = gpcd_create_context();
for (i = 0; i < 6; i++)  // TILE_ROWS
    for (j = 0; j < 8; j++)  // TILE_COLS
        for (k = 0; k < 6; k++)  // TILE_COUNTERS
            sprintf(name,
                "GM%d%d_TILE_PERFORMANCE_COUNTERS%d",
                i, j, k);
        desc = gpcd_lookup_mmr_bname(name);
        gpcd_context_add_mmr(ctx, desc);
}

// Sample the tile counters
gpcd_context_read_mmr_vals(ctx);

// Print the counter values
for (p = ctx->list; p; p = p->next)
    printf("Counter %s: Value=%lu\n",
        p->item->name, p->value);
```

- GPCD Library available in Gemini Kernel driver source code (GPLv2)
- Code sets up to sample the 288 tile counters, 48 tiles * 6 counters
- Traps to kernel to read counters, driver ioctl()
- Benchmark, time to sample all 288 tile counters:
  - Average: 159 us
  - Min: 154 us
  - Max: 305 us
- => Slow Operation! Use with care
Aggregating to Link Counters

- Tile counters are just an implementation detail
  - Really care about the logical network links
  - Need to figure out which tiles make up each network link

- No obvious way to get the mapping from compute nodes
  - Instead, use Cray’s rtr tool available on the SMW to dump map
  - Our tools depend on this text file, parse at startup

```
  rtr --interconnect > interconnect.txt
```

<table>
<thead>
<tr>
<th>Source Gemini Tile</th>
<th>Destination Gemini Tile</th>
<th>Type of Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>c0-0c0s0g0100[(0,0,0)] Z+</td>
<td>c0-0c0s1g0132[(0,0,1)]</td>
<td>LinkType: backplane</td>
</tr>
<tr>
<td>c0-0c0s0g0101[(0,0,0)] Z+</td>
<td>c0-0c0s1g0121[(0,0,1)]</td>
<td>LinkType: backplane</td>
</tr>
<tr>
<td>c0-0c0s0g0102[(0,0,0)] X+</td>
<td>c1-0c0s0g0102[(1,0,0)]</td>
<td>LinkType: cable11x</td>
</tr>
<tr>
<td>c0-0c0s0g0103[(0,0,0)] X+</td>
<td>c1-0c0s0g0103[(1,0,0)]</td>
<td>LinkType: cable11x</td>
</tr>
<tr>
<td>c0-0c0s0g0104[(0,0,0)] X-</td>
<td>c2-0c0s0g0141[(15,0,0)]</td>
<td>LinkType: cable18x</td>
</tr>
<tr>
<td>c0-0c0s0g0105[(0,0,0)] X-</td>
<td>c2-0c0s0g0131[(15,0,0)]</td>
<td>LinkType: cable18x</td>
</tr>
<tr>
<td>c0-0c0s0g0106[(0,0,0)] Z-</td>
<td>c0-0c2s7g0126[(0,0,23)]</td>
<td>LinkType: cable15z</td>
</tr>
<tr>
<td>c0-0c0s0g0107[(0,0,0)] Z-</td>
<td>c0-0c2s7g0135[(0,0,23)]</td>
<td>LinkType: cable15z</td>
</tr>
</tbody>
</table>
Example Tile to Logical Link Mapping
For LANL/SNL Cielo Cray XE6

<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:**
\[8 \times 1.88 = 15 \text{ GB/s (backpl)}\]
\[8 \times 1.17 = 9.4 \text{ GB/s}\]
Gathering Job Wide Information

// Initialize the library
gemini_init_state(comm, &state)

// Sample the gemini counters
gemini_read_counters(comm, &state)

// Output delta of last two samples
gemini_print_counters(comm, &state)

### Table II: Tile link type to bandwidth conversions.

<table>
<thead>
<tr>
<th>DEST_COORD</th>
<th>SRC_COORD</th>
<th>GB/s</th>
<th>VC0_PHITS</th>
<th>VC1_PHITS</th>
<th>VC0_PKTS</th>
<th>VC1_PKTS</th>
<th>INQ_STALLS</th>
<th>OUTQ_STALLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 1, 1)</td>
<td>X+</td>
<td>9.38</td>
<td>1626452284</td>
<td>304999266</td>
<td>101662806</td>
<td>101666422</td>
<td>3533598201</td>
<td>2689080952</td>
</tr>
<tr>
<td></td>
<td>X-</td>
<td>9.38</td>
<td>100506</td>
<td>38796</td>
<td>9780</td>
<td>12932</td>
<td>83</td>
<td>0</td>
</tr>
<tr>
<td>(0, 2, 1)</td>
<td>Y+</td>
<td>4.69</td>
<td>1627257610</td>
<td>305156760</td>
<td>101726643</td>
<td>10171920</td>
<td>1702270109</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Y-</td>
<td>9.38</td>
<td>115354135</td>
<td>21631236</td>
<td>72105559</td>
<td>72104412</td>
<td>1925883229</td>
<td>2366983378</td>
</tr>
<tr>
<td>(0, 1, 2)</td>
<td>Z+</td>
<td>15.00</td>
<td>815234359</td>
<td>152948952</td>
<td>50988260</td>
<td>50982984</td>
<td>133047991</td>
<td>776502961</td>
</tr>
<tr>
<td></td>
<td>Z-</td>
<td>15.00</td>
<td>1743043</td>
<td>378399</td>
<td>156635</td>
<td>126133</td>
<td>580</td>
<td>992022669</td>
</tr>
<tr>
<td>(0, 1, 0)</td>
<td>HH</td>
<td>10.40</td>
<td>1834489368</td>
<td>344019696</td>
<td>114672167</td>
<td>114673232</td>
<td>10585723107</td>
<td>2263990777</td>
</tr>
<tr>
<td>(0, 0, 1)</td>
<td>X+</td>
<td>9.38</td>
<td>1966685020</td>
<td>368797209</td>
<td>122929393</td>
<td>122932403</td>
<td>3317929506</td>
<td>3063532486</td>
</tr>
<tr>
<td></td>
<td>X-</td>
<td>9.38</td>
<td>122194</td>
<td>43005</td>
<td>11983</td>
<td>14335</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>(0, 1, 1)</td>
<td>Y+</td>
<td>9.38</td>
<td>1154016206</td>
<td>216417552</td>
<td>72141025</td>
<td>72139184</td>
<td>3589170400</td>
<td>1097189607</td>
</tr>
<tr>
<td></td>
<td>Y-</td>
<td>4.69</td>
<td>96911</td>
<td>20538</td>
<td>9646</td>
<td>6846</td>
<td>56244</td>
<td>0</td>
</tr>
<tr>
<td>(0, 2, 0)</td>
<td>Z+</td>
<td>15.00</td>
<td>2477453033</td>
<td>458007486</td>
<td>153779007</td>
<td>152669162</td>
<td>952487628</td>
<td>2209098748</td>
</tr>
<tr>
<td></td>
<td>Z-</td>
<td>15.00</td>
<td>2071415</td>
<td>3684912</td>
<td>128723</td>
<td>1228304</td>
<td>464902</td>
<td>387186094</td>
</tr>
<tr>
<td>(0, 0, 1)</td>
<td>HH</td>
<td>10.40</td>
<td>2174662127</td>
<td>407809092</td>
<td>135934105</td>
<td>135936364</td>
<td>10604254673</td>
<td>2216827070</td>
</tr>
</tbody>
</table>
Outline

- Motivation and background
- How to access

What they measure

- Usage example: MiniGhost rank remapping
- Conclusion
“Sonar” Experiments

- Basic Idea: Send out a known ping, observe tile counters to figure out what happened
- First test, send a 1 MB MPI message between two nodes
- Expect either PUT or GET transactions
  - Both transaction types move up to 64 bytes of user data
  - PUT transactions consist of 32 phit (96 byte) request packet on VC0 followed by a 3 phit (9 byte) response packet on VC1 (the ACK)
  - GET transactions consist of 8 phit (24 byte) request packet on VC0 followed by a 27 phit (81 byte) response packet on VC1 (the REPLY)
1 MB Point-to-Point MPI Test

- Original counts from sender’s perspective:
  - Packets: TX = 16,407 RX = 16,407 (Expected 16,384)
  - Phits: TX = 262,565 RX = 49,221 (Expected TX 524,288 RX 49,152)

- Packet counters make sense, phit counters too low by 2x

- After discussing with Cray, due to compression (!)
  - Our test was sending a zero’ed message
  - The Gemini compresses runs of zeros and ones

- After initializing buffer to random bits, phit counters made sense 😊
  - Phits: TX = 524,709 RX = 49,221
# Tile Counter Directionality (1)

- Packet and Phit counters measure input into tile, not output.

```plaintext
src=0, dst=2
```

<table>
<thead>
<tr>
<th>Location</th>
<th>Direction</th>
<th>Value</th>
<th>Phit Count</th>
<th>Packets</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 0, 5)</td>
<td>SOURCE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X- (0, 0, 5)</td>
<td>18.75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y+ (1, 1, 5)</td>
<td>14.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Z+ (1, 0, 6)</td>
<td>15.00</td>
<td>278</td>
<td>49203</td>
<td>14</td>
<td>16401</td>
<td>0</td>
</tr>
<tr>
<td>Z- (1, 0, 4)</td>
<td>15.00</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HH (1, 0, 5)</td>
<td>10.40</td>
<td>524580</td>
<td>42</td>
<td>16401</td>
<td>14</td>
<td>817252</td>
</tr>
<tr>
<td>(1, 0, 6)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X- (0, 0, 6)</td>
<td>18.75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y+ (1, 1, 6)</td>
<td>14.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Z+ (1, 0, 7)</td>
<td>15.00</td>
<td>278</td>
<td>49203</td>
<td>14</td>
<td>16401</td>
<td>0</td>
</tr>
<tr>
<td>Z- (1, 0, 5)</td>
<td>15.00</td>
<td>524580</td>
<td>42</td>
<td>16401</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>HH (1, 0, 6)</td>
<td>10.40</td>
<td>156</td>
<td>24</td>
<td>8</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>(1, 0, 7)</td>
<td>DESTINATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X- (0, 0, 7)</td>
<td>18.75</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Y+ (1, 1, 7)</td>
<td>14.06</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Z+ (1, 0, 0)</td>
<td>9.38</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Z- (1, 0, 6)</td>
<td>15.00</td>
<td>524584</td>
<td>42</td>
<td>16401</td>
<td>14</td>
<td>57659</td>
</tr>
<tr>
<td>HH (1, 0, 7)</td>
<td>10.40</td>
<td>278</td>
<td>49203</td>
<td>14</td>
<td>16401</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Tile Counter Directionality (2)

- Graphical view of data on previous slide

Source Host

Gemini (1, 0, 5)

524K HH

49K Z+

Gemini (1, 0, 6)

524K Z-

49K Z+

Gemini (1, 0, 7)

49K HH

524K Z-
Routing

- 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
The maximum congestion on any link is the bottleneck for performance. For X dimension output stalls, the average stall count actually increases slightly for the MiniGhost, since it is a bulk-synchronous parallel application. The maximum congestion on any link is the bottleneck for performance counters to quantify the reduction in network messaging. Finally, we demonstrated the use of the Gemini’s network processing unit. This included a description of how to directly access the counters from a user application, of the Gemini tile counters, including their directionality, and the results of several experiments designed to understand the operation of the system as a whole in terms of network transactions, routing, and MPI point-to-point messaging. We plan to perform similar studies for other applications and task mapping algorithms in the future.

For all dimensions (X, Y, and Z), the added contention with traffic waiting to move into the Y and Z dimensions. For all Geminis in a run, the error bars represent the average per-Gemini remote communication time. The plots show that X dimension network links encounter the highest maximum congestion. We believe this is due to the dynamic task mapping based on network congestion. Specifically, we would like to make use of MPI scalable task mapping strategies with respect to network congestion. In this paper we have described our method for accessing performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quantify the benefit of different performance counters and have tools for analyzing them, we plan to use the capability to quanti...
Outline

- Motivation and background
- How to access
- What they measure

Usage example: MiniGhost rank remapping

- Conclusion
Large-scale MiniGhost Experiments

- 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
Reducing Off-node Communication

- Changing the mapping of MPI processes to nodes affects off-node communication
- Used Gemini tile counters to measure traffic injected on the host links
- The reordered "Remap" scheme (2x2x4) reduces off-node communication by more than a factor of 2x compared to the original "No-Remap" scheme (1x1x16)
Stalls Correlate with Communication Time

**Per-Gemini Input Stalls**

- Processes (16 Processes/Node)
- Input Queue Stall Cycles
- Random, No-Remap, Remap

**Per-Rank Communication Time**

- Processes (16 Processes/Node)
- Seconds
- Random, No-Remap, Remap

---

22
Per-Link Input and Output Stalls
128K Process Runs (4K Geminis)

- Remap scheme reduces maximum load on any link (error bars)
- X-dimension has highest congestion, likely due to routing alg.

<table>
<thead>
<tr>
<th>Input Stalls</th>
<th>Output Stalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link Type</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
</tr>
<tr>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>Link Type</td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
</tr>
<tr>
<td>Host</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td></td>
</tr>
</tbody>
</table>

- Input Queue Stall Cycles
- Output Queue Stall Cycles

<table>
<thead>
<tr>
<th>Link Type</th>
<th>Input Stalls</th>
<th>Output Stalls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>1.6e+11</td>
<td>1.6e+11</td>
</tr>
<tr>
<td>Host</td>
<td>1.4e+11</td>
<td>1.4e+11</td>
</tr>
<tr>
<td>X</td>
<td>1.2e+11</td>
<td>1.2e+11</td>
</tr>
<tr>
<td>Y</td>
<td>1.0e+11</td>
<td>1.0e+11</td>
</tr>
<tr>
<td>Z</td>
<td>8.0e+10</td>
<td>8.0e+10</td>
</tr>
</tbody>
</table>

- Random
- No-Remap
- Remap

[Graphs showing input and output stall cycles for different link types.]
Outline

- Motivation and background
- How to access
- What they measure
- Usage example: MiniGhost rank remapping

Conclusion
Future Work

- Investigate Cray PAPI support for Gemini and Aries

- Evaluating topology mapping strategies
- Dynamic (re)partitioning based on real-time counter info
- Investigate Aries network
Conclusion

- Direct access to Gemini tile performance counters
- Convert tile counters to logical network link counters
- Gemini counter operation
  - Put and Get transactions
  - Counter directionality (count incoming packets/phits)
  - Routing
  - MPI bandwidth efficiency
- Used counters to quantify MiniGhost rank remapping scheme
- Plan to release Gemini Monitor library as open source, email ktpedre@sandia.gov in the meantime
Backup Slides
MPI Point-to-Point BW Efficiency (Overall, Transmit + Receive)

- This plot includes all traffic on the wire, including response traffic (PUT ACK and GET REPLY)
- Large messages achieve ~60% bandwidth efficiency; larger max packet size would help