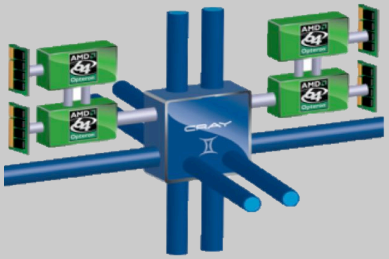
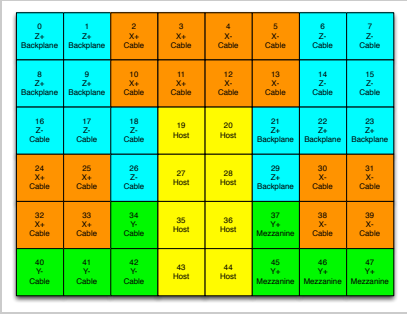




Sandia National Laboratories

Exceptional service in the national interest

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

U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Outline

Sandia National Laboratories

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

Motivation

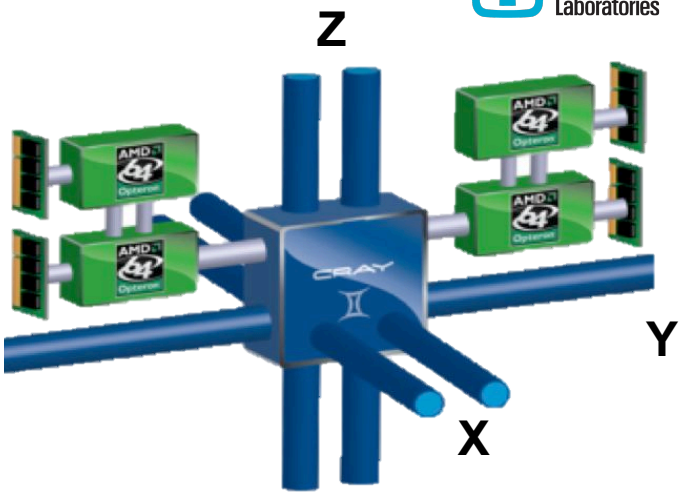
Sandia National Laboratories

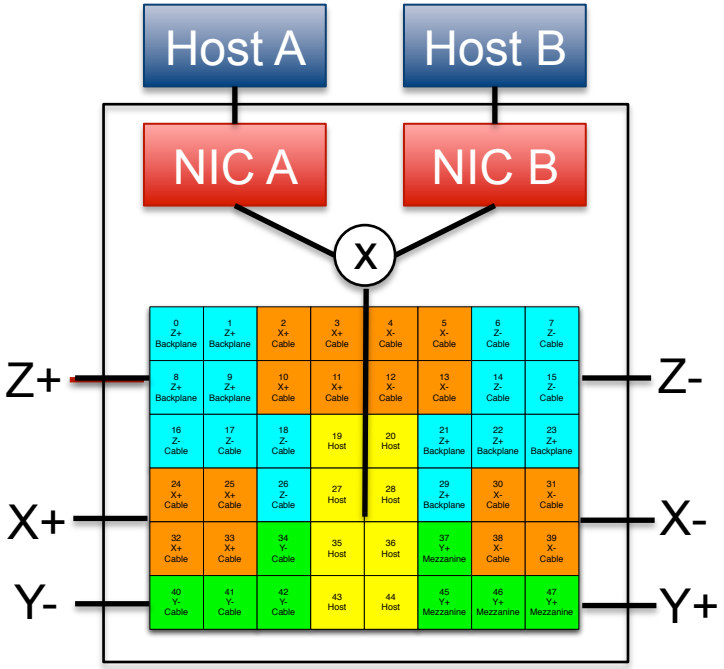
- We had an application that was scaling well to 16K processes, then poorly afterwards (weak scaling)
- We suspected network congestion/contention was becoming an issue and wanted to quantify it empirically
- We had heard the Gemini had a nice set of performance counters that could do this

➜ 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

Sandia National Laboratories

- 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?

Sandia National Laboratories

- 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

Sandia National Laboratories

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

Directly Accessing Gemini Counters

Sandia National Laboratories

```
1  gpcd_context_t *ctx;
2  gpcd_mmr_desc_t *desc;
3  gpcd_mmr_list_t *p;
4  int i, j, k;
5  char name[128];
6
7  // Create a counter group, all tile counters
8  ctx = gpcd_create_context();
9  for (i = 0; i < 6; i++)       // TILE_ROWS
10   for (j = 0; j < 8; j++)     // TILE_COLS
11     for (k = 0; k < 6; k++) // TILE_COUNTERS
12     {
13       sprintf(name,
14         "GM_%d_%d_TILE_PERFORMANCE_COUNTERS_%d",
15         i, j, k);
16       desc = gpcd_lookup_mmr_byname(name);
17       gpcd_context_add_mmr(ctx, desc);
18     }
19
20 // Sample the tile counters
21 gpcd_context_read_mmr_vals(ctx);
22
23 // Print the counter values
24 for (p = ctx->list; p; p = p->next)
25   printf("Counter %s: Value=%lu\n",
26          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

Sandia National Laboratories

- 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
```

Source Gemini Tile		Destination Gemini Tile	Type of Link
c0-0c0s0g0l00[(0,0,0)]	Z+ ->	c0-0c0s1g0l32[(0,0,1)]	LinkType: backplane
c0-0c0s0g0l01[(0,0,0)]	Z+ ->	c0-0c0s1g0l21[(0,0,1)]	LinkType: backplane
c0-0c0s0g0l02[(0,0,0)]	X+ ->	c1-0c0s0g0l02[(1,0,0)]	LinkType: cable11x
c0-0c0s0g0l03[(0,0,0)]	X+ ->	c1-0c0s0g0l03[(1,0,0)]	LinkType: cable11x
c0-0c0s0g0l04[(0,0,0)]	X- ->	c2-0c0s0g0l41[(15,0,0)]	LinkType: cable18x
c0-0c0s0g0l05[(0,0,0)]	X- ->	c2-0c0s0g0l31[(15,0,0)]	LinkType: cable18x
c0-0c0s0g0l06[(0,0,0)]	Z- ->	c0-0c2s7g0l26[(0,0,23)]	LinkType: cable15z
c0-0c0s0g0l07[(0,0,0)]	Z- ->	c0-0c2s7g0l35[(0,0,23)]	LinkType: cable15z

Example Tile to Logical Link Mapping For LANL/SNL Cielo Cray XE6

Sandia National Laboratories

0 Z+ Backplane	1 Z+ Backplane	2 X+ Cable	3 X+ Cable	4 X- Cable	5 X- Cable	6 Z- Cable	7 Z- Cable
8 Z+ Backplane	9 Z+ Backplane	10 X+ Cable	11 X+ Cable	12 X- Cable	13 X- Cable	14 Z- Cable	15 Z- Cable
16 Z- Cable	17 Z- Cable	18 Z- Cable	19 Host	20 Host	21 Z+ Backplane	22 Z+ Backplane	23 Z+ Backplane
24 X+ Cable	25 X+ Cable	26 Z- Cable	27 Host	28 Host	29 Z+ Backplane	30 X- Cable	31 X- Cable
32 X+ Cable	33 X+ Cable	34 Y- Cable	35 Host	36 Host	37 Y+ Mezzanine	38 X- Cable	39 X- Cable
40 Y- Cable	41 Y- Cable	42 Y- Cable	43 Host	44 Host	45 Y+ Mezzanine	46 Y+ Mezzanine	47 Y+ Mezzanine

Link Type	Bandwidth
Mezzanine	2.34 GB/s
Backplane	1.88 GB/s
Cable	1.17 GB/s
Host	1.33 GB/s (est.)

Unidirectional Bandwidths

X Links, all:

8 * 1.17 = 9.4 GB/s

Y Links, alternate every other:

4 * 2.34 = 9.4 GB/s (mezz)

4 * 1.17 = 4.7 GB/s

Z Links, every eighth slower:

8 * 1.88 = 15 GB/s (backpl)

8 * 1.17 = 9.4 GB/s

Gathering Job Wide Information

Sandia National Laboratories

```
// 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)
```

```
# DEST_COORD
#    SRC_COORD  GB/s   VC0_PHITS  VC1_PHITS VC0_PKTS  VC1_PKTS   INQ_STALLS OUTQ_STALLS

( 0, 1, 1 )
 X+ ( 1, 1, 1)  9.38   1626452284 304999266 101662806 101666422  3533598961  2689080952
 X- (15, 1, 1)  9.38       100506     38796      9780     12932          83           0
 Y+ ( 0, 2, 1)  4.69   1627257610 305156760 101726643 101718920  1702270109           0
 Y- ( 0, 0, 1)  9.38   1153554135 216313236  72105559  72104412  1925883229  2366983378
 Z+ ( 0, 1, 2) 15.00    815234359 152948952  50988260  50982984   133047991   776502961
 Z- ( 0, 1, 0) 15.00      1743043    378399    156635    126133         580   992022669
 HH ( 0, 1, 1) 10.40   1834489368 344019696 114672167 114673232 10585723107  2263990777

( 0, 0, 1 )
 X+ ( 1, 0, 1)  9.38   1966685020 368797209 122929393 122932403  3317929506  3063532486
 X- (15, 0, 1)  9.38       122194     43005     11983     14335           9           0
 Y+ ( 0, 1, 1)  9.38   1154016206 216417552  72141025  72139184  3589170400  1097189607
 Y- ( 0,11, 1)  4.69        96911     20538      9646      6846       56244           0
 Z+ ( 0, 0, 2) 15.00   2477453033 458007486 153779007 152669162   952487628  2209098748
 Z- ( 0, 0, 0) 15.00      2071415   3684912    128723   1228304      464902   387186094
 HH ( 0, 0, 1) 10.40   2174662127 407809092 135934105 135936364 10604254673  2216827070
```

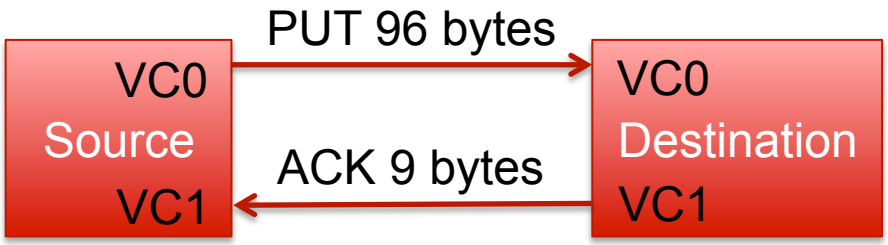
Outline

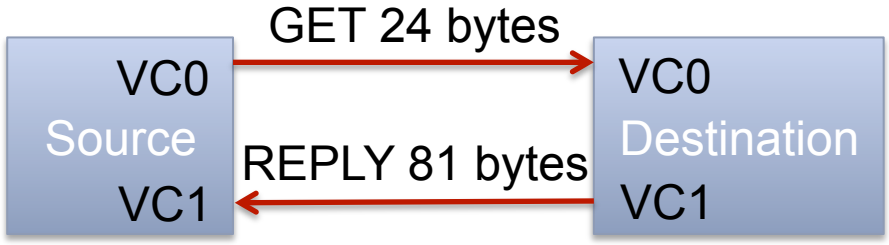
Sandia National Laboratories

- 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

Sandia National Laboratories

- 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)

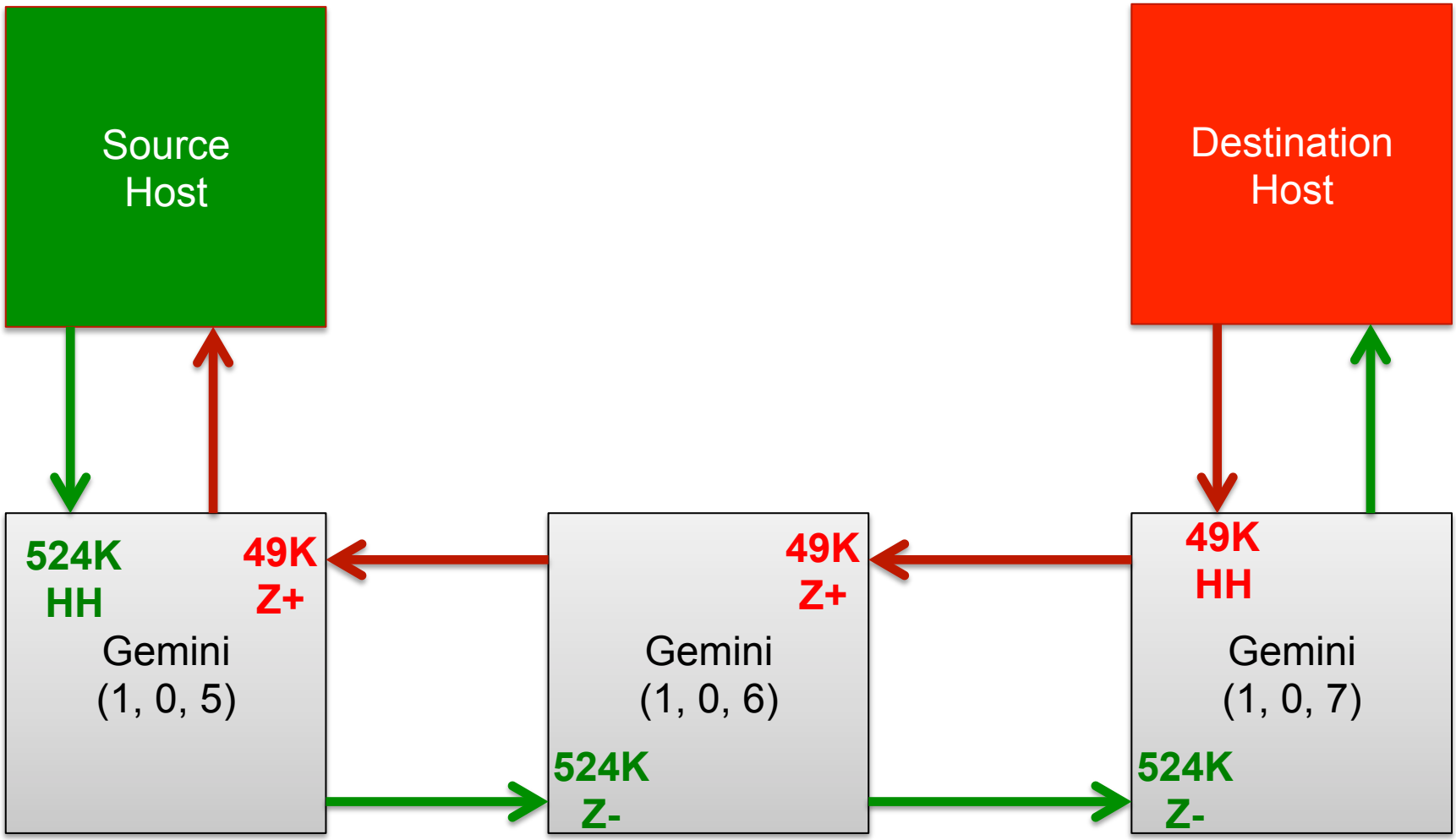
Sandia National Laboratories

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

```
src=0, dst=2
================================================================================
(1, 0, 5) SOURCE
  X- (0, 0, 5) 18.75            0            0            0            0            0            0
  Y+ (1, 1, 5) 14.06            0            0            0            0            0            0
  Z+ (1, 0, 6) 15.00          278        49203           14        16401            0       141274
  Z- (1, 0, 4) 15.00            0            0            0            0            0            0
  HH (1, 0, 5) 10.40       524580           42        16401           14       817252            0
(1, 0, 6)
  X- (0, 0, 6) 18.75            0            0            0            0            0            0
  Y+ (1, 1, 6) 14.06            0            0            0            0            0            0
  Z+ (1, 0, 7) 15.00          278        49203           14        16401            0            0
  Z- (1, 0, 5) 15.00       524580           42        16401           14            0            0
  HH (1, 0, 6) 10.40          156           24            8            8            0            0
(1, 0, 7) DESTINATION
  X- (0, 0, 7) 18.75            0            0            0            0            0            0
  Y+ (1, 1, 7) 14.06            0            0            0            0            0            0
  Z+ (1, 0, 0)  9.38            0            0            0            0            0            0
  Z- (1, 0, 6) 15.00       524584           42        16401           14        57659            0
  HH (1, 0, 7) 10.40          278        49203           14        16401            0            0
```

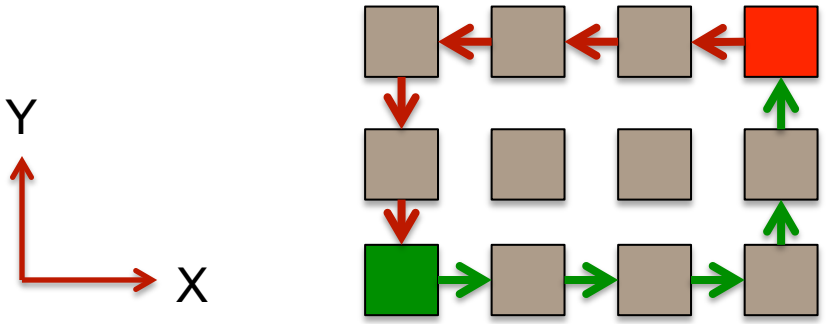
Tile Counter Directionality (2)

- Graphical view of data on previous slide

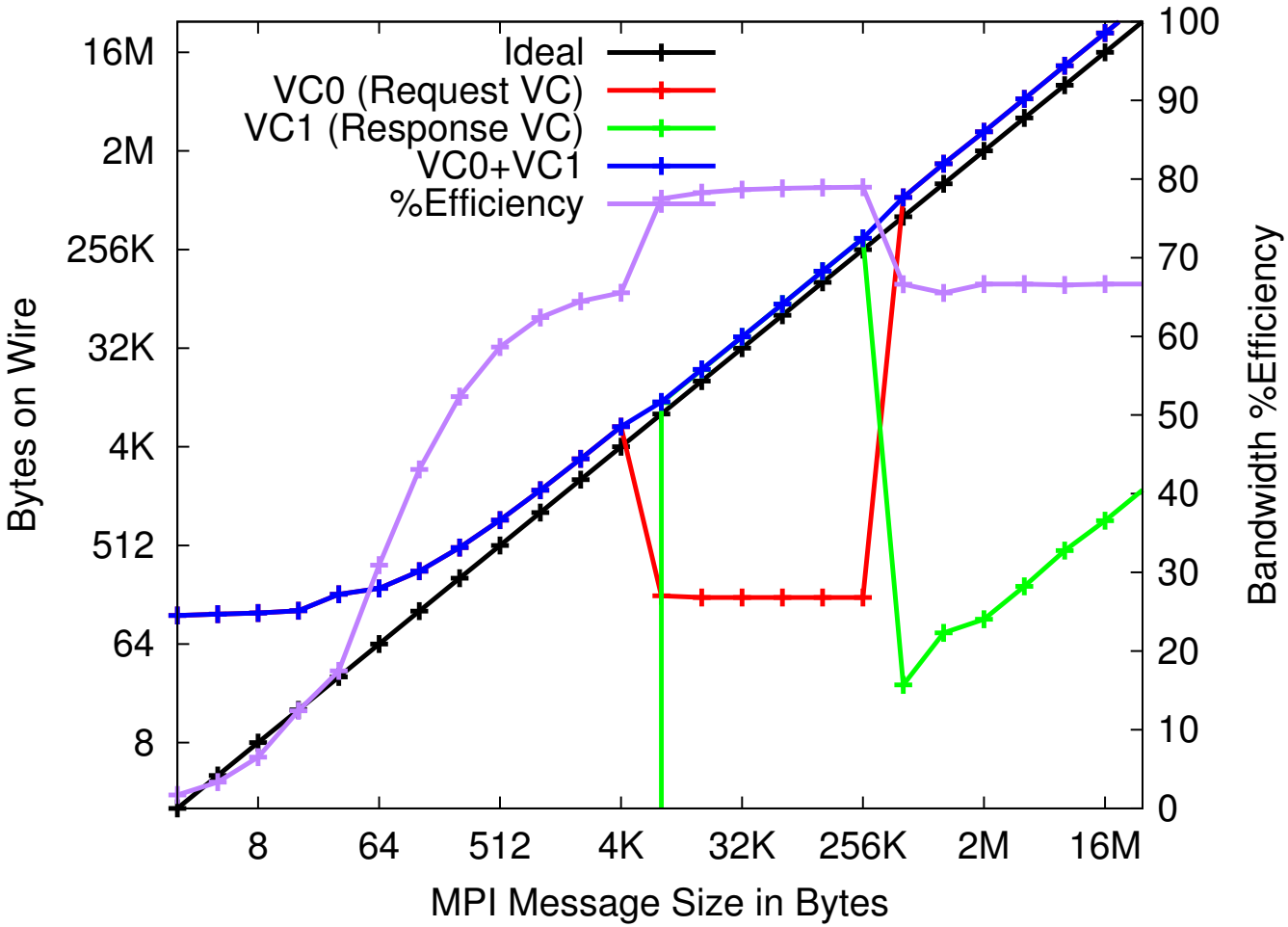


Routing

- 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



MPI Point-to-Point BW Efficiency (Initiator Transmit Only)



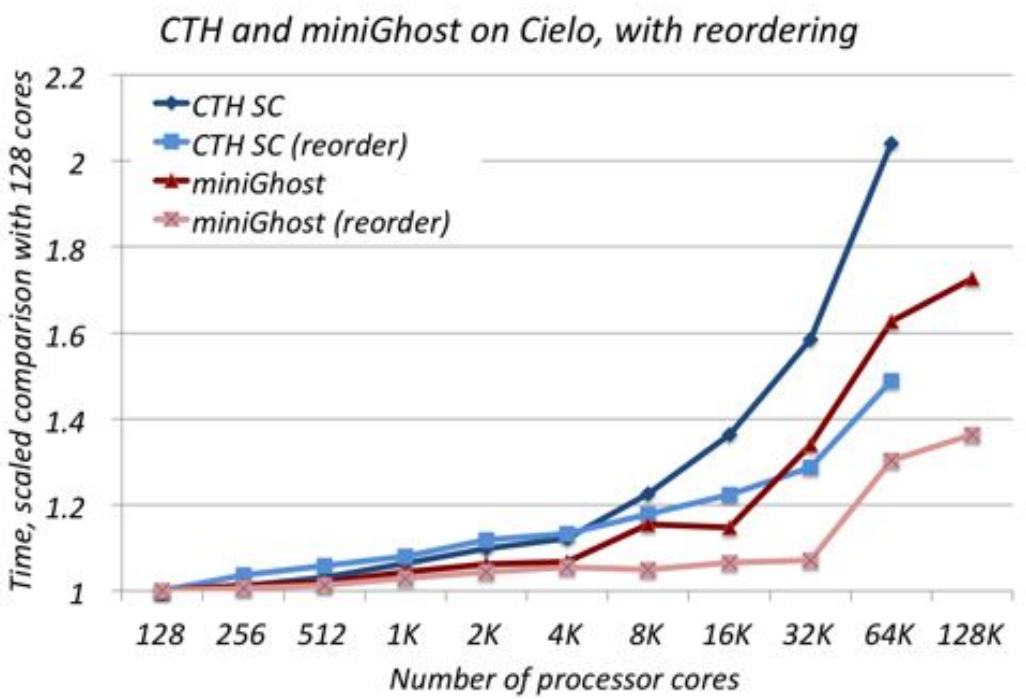
- Observing VC0 and VC1 traffic reveals the MPI protocols, PUT push vs. GET pull based
- Purple curve plots bandwidth efficiency, uses scale on right Y axis
- Lower bandwidth efficiency for small messages, due to ~64 byte MPI header
- Results highly repeatable, appear accurate even for zero byte messages

Outline

Sandia National Laboratories

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

Large-scale MiniGhost Experiments

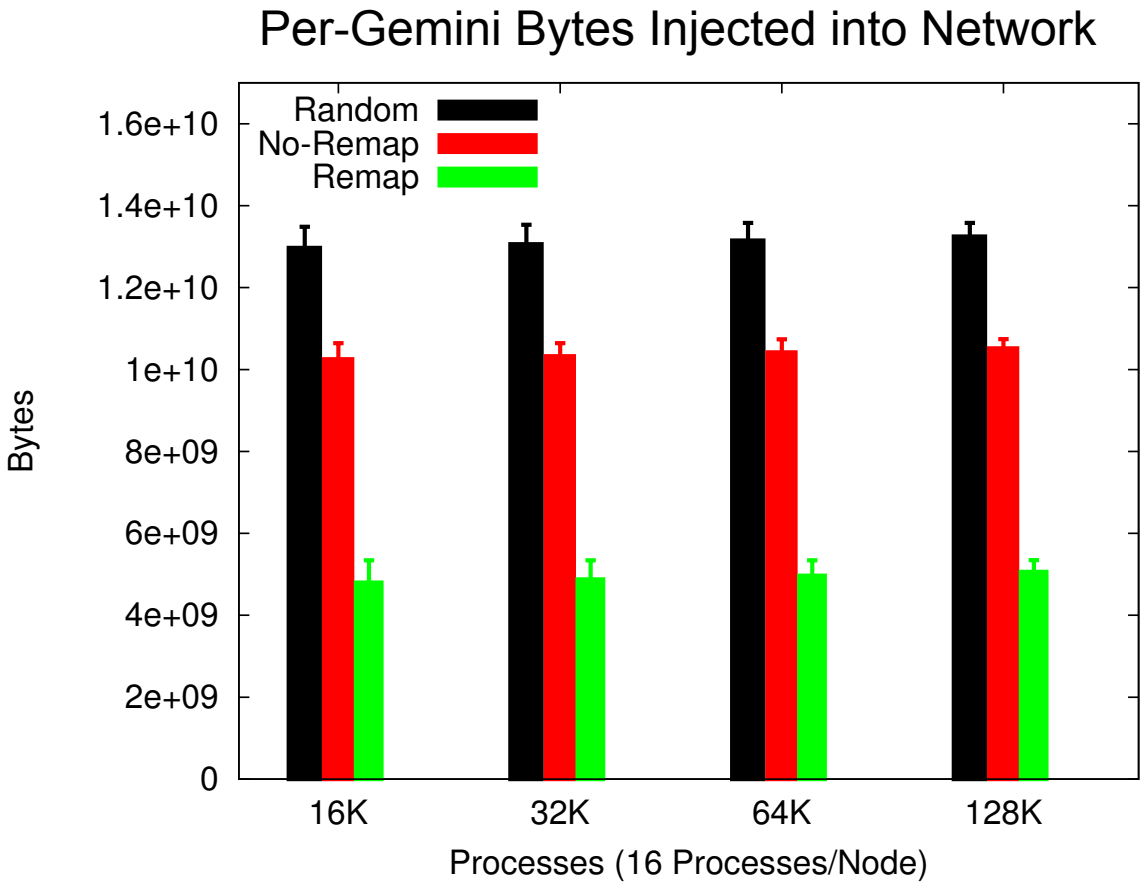


- MiniGhost is a proxy application, represents CTH full application
- Explicit time-stepping, synchronous communication, 27-point stencil across 3-D grid
- Dark Red Curve:
  Original configuration
  scaled poorly after 16K cores
  (1024 nodes, 512 Geminis)
- Light Red Curve:
  Reorder MPI rank to node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

  Original: 1x1x16 ranks/node

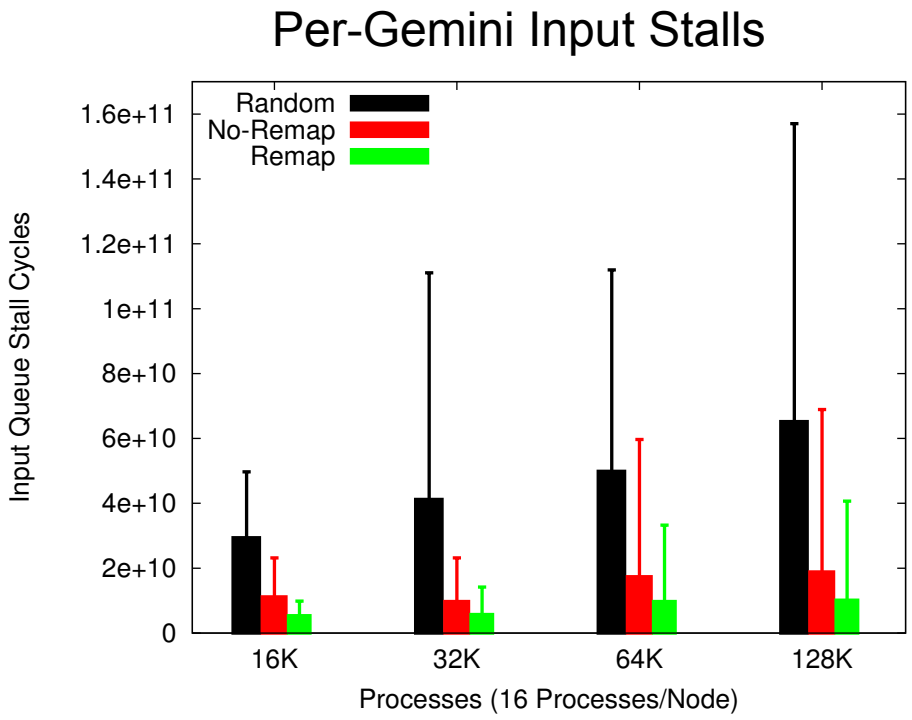
  Reorder: 2x2x4 ranks/node

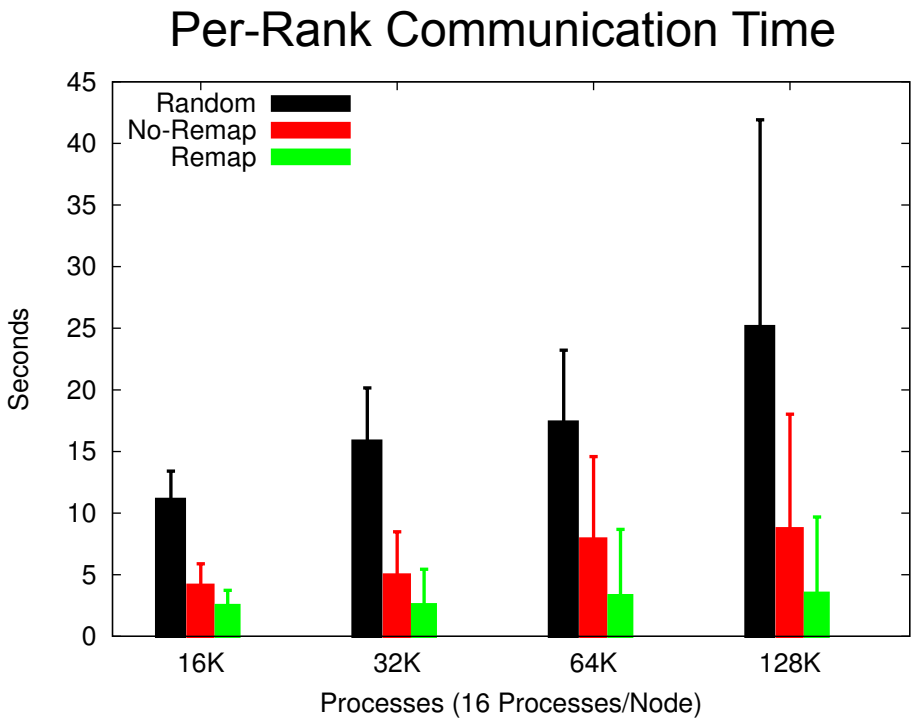
Reducing Off-node Communication



- 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

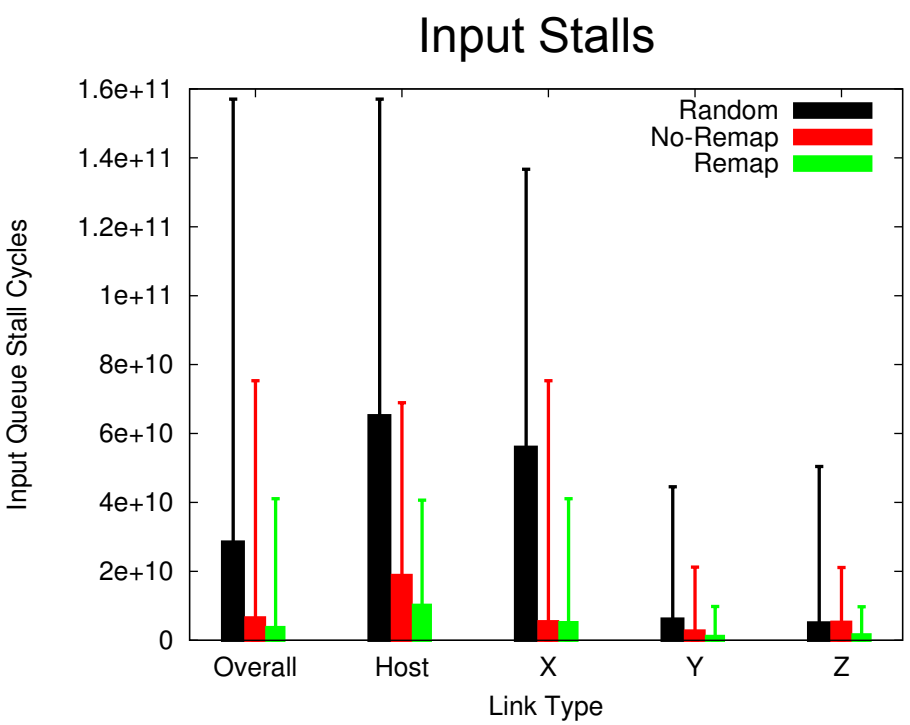


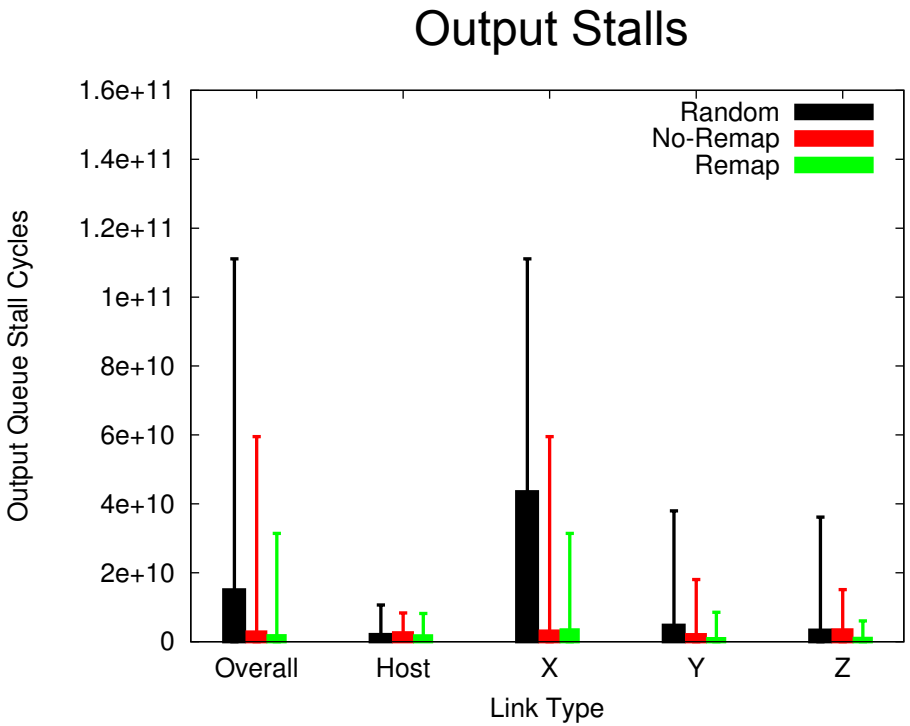


Per-Link Input and Output Stalls 128K Process Runs (4K Geminis)

Sandia National Laboratories

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





Outline

Sandia National Laboratories

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

Future Work

Sandia National Laboratories

- Investigate Cray PAPI support for Gemini and Aries
  - *Using the PAPI Cray NPU Component,* Cray Inc., Mar. 2013. Available: http://docs.cray.com/books/S-0046-10/

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

Conclusion

Sandia National Laboratories

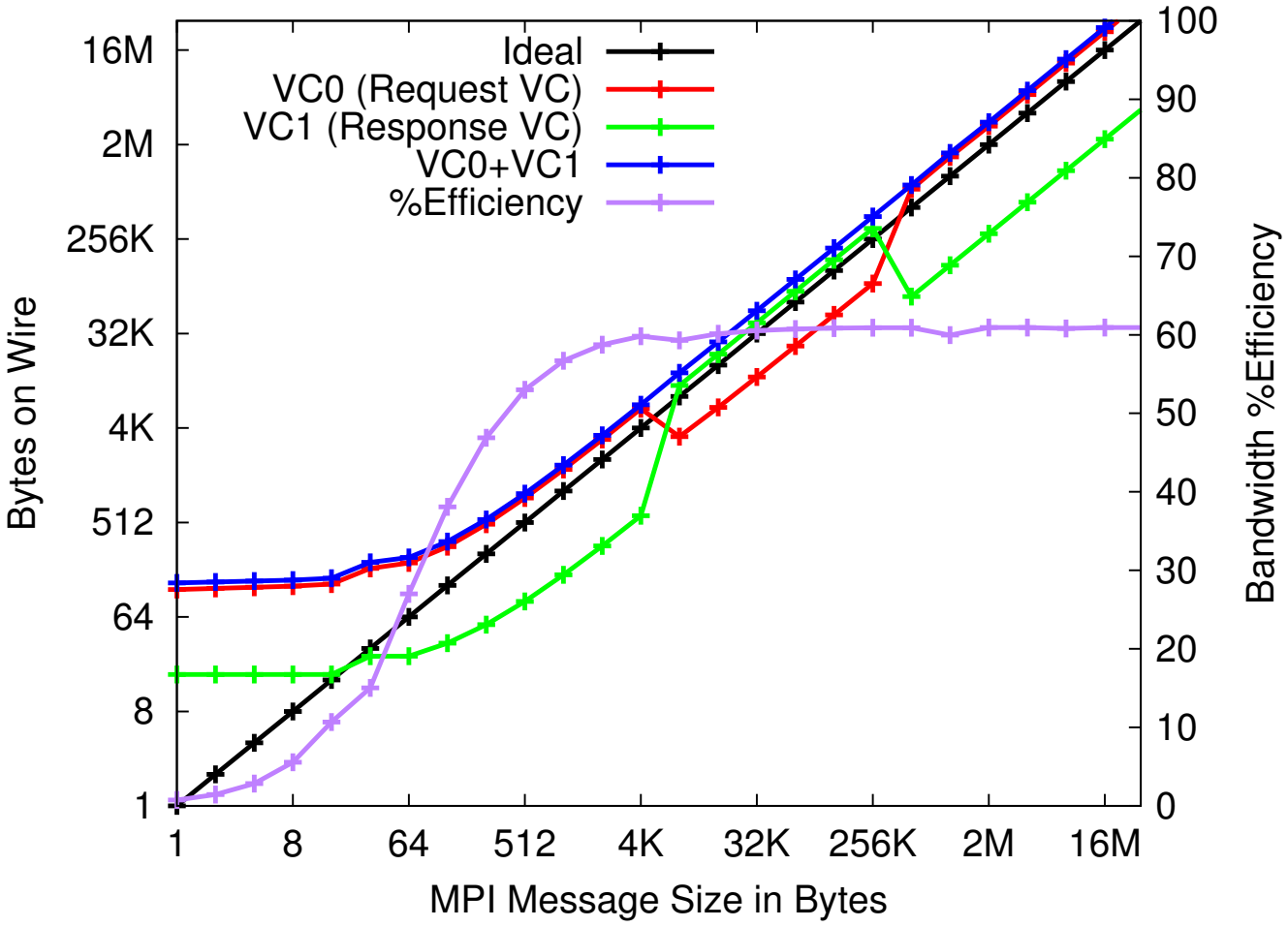
- Direct access to Gemini tile performance counters
- Convert tile counters to logical network link counters
- Gemini counter operation
  - Put and Get transactions
  - Counter directionality (count incoming packets/phits)
  - Routing
  - MPI bandwidth efficiency
- Used counters to quantify MiniGhost rank remapping scheme

- Plan to release Gemini Monitor library as open source, email ktpedre@sandia.gov in the meantime

Sandia National Laboratories

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