OAK RIDGE National Laboratory

Summit Interconnection Network Spectrum MPI and InfiniBand

Christopher Zimmer

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

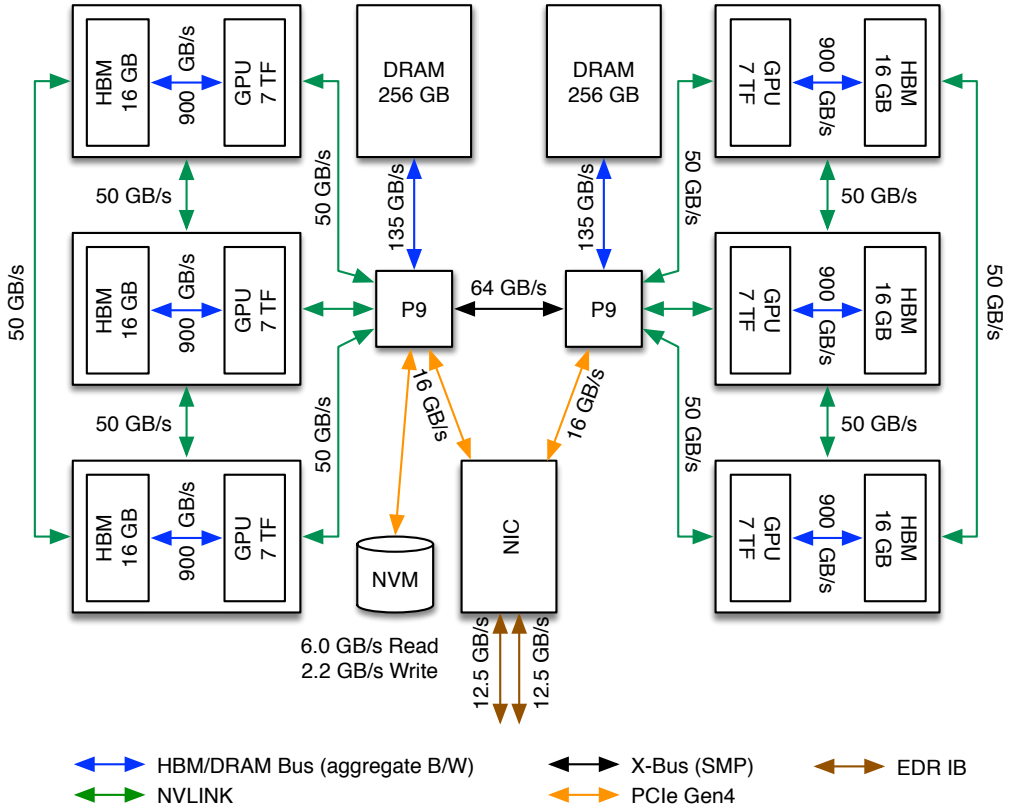
U.S. DEPARTMENT OF ENERGY

Summit Topology

- Non-blocking Fat Tree
  - No global bandwidth reduction
- Global communication
  - Worst case 5 switch hops
  - Latency reduction over Titan (25x16x24) (9600 Switches)
- 18 Rack Neighborhoods
  - Jobs less than 18 nodes should request single CU packing for network packing
  - Smaller jobs (less than 18 racks) with tight CU compaction will have a 3 hop subtree

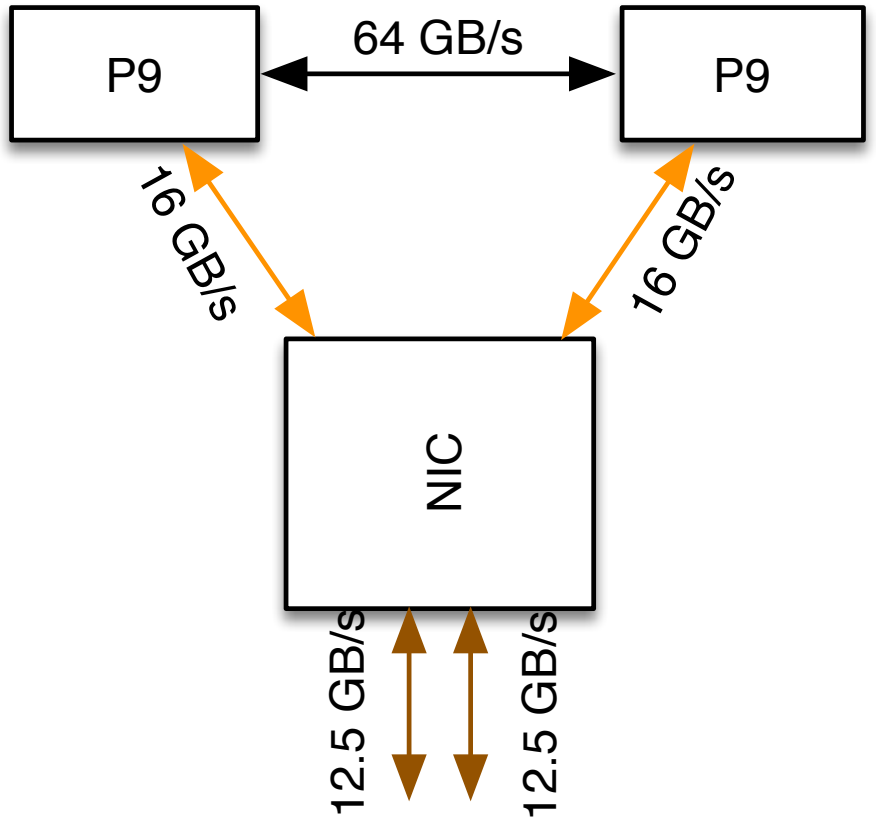
Summit Node

- 100 Gbps EDR InfiniBand
- 2 Physical Ports
- 4 Virtual Ports
  - MLX5_0 MLX5_1 (Socket 0)
  - MLX5_2 MLX5_3 (Socket 1)



Some oddities

- 16 GB/s of PCI-Gen 4 per socket into NIC
  - To get full HCA bandwidth 25 GB/s must use both sockets



Controlling Port Use via Spectrum MPI

- PAMI_IBV_ADAPTER_AFFINITY=1 (Default) 0 off

- PAMI_IBV_DEVICE_NAME=name:port Primary socket
  - **PAMI_IBV_DEVICE_NAME="mlx5_0:1,mlx5_1:1"**
- PAMI_IBV_DEVICE_NAME_1=name:port Ports for remote socket (Only used by remote socket if set)
  - **PAMI_IBV_DEVICE_NAME_1="mlx5_2:1,mlx5_3:1"**

Striping Across Multiple Ports

- **PAMI_ENABLE_STRIPING=1**
  - Will increase per rank bandwidth
  - Striping only used for RDMA messages
  - May increase latency

Dynamic Connected Queue Pairs

- **PAMI_IBV_ENABLE_DCT=1**
  - Speed up initialization (MPI_Init)
  - Use at large scale

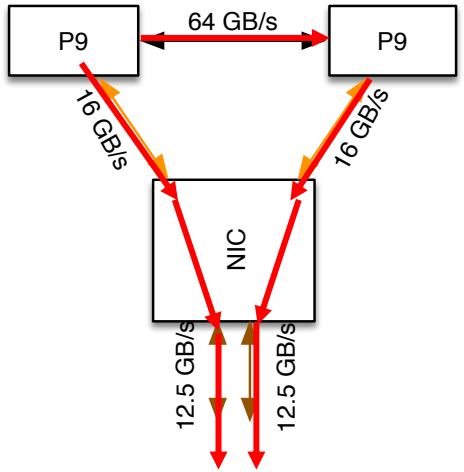
- Can be some impacts to performance (marginal)

Adaptive Routing

- Infiniband has historically been statically routed
  - This means no bypassing congestion
- Summit ConnectX-5 + Switch IB 2 supports adaptive routing
  - On by default
- Enables more effective capture of bandwidth and reduces tail latency
- On by default (There should be very little reason to turn this off)
  - BUT if you feel it could be causing issues (Disabling this can impact other jobs by increasing congestion so tread carefully)
    - **PAMI_IBV_ENABLE_OOO_AR=1**

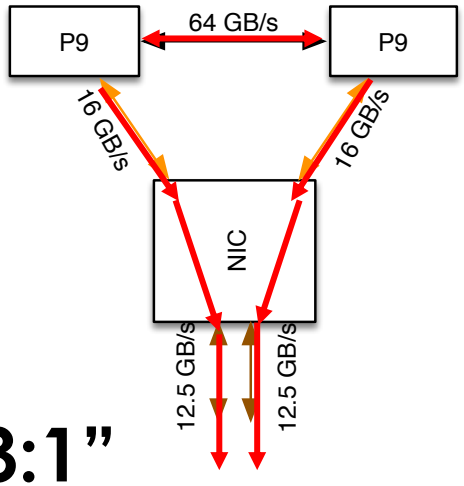
Example Configurations – Single Rank Full Bandwidth

- export PAMI_IBV_ENABLE_DCT=1
- export PAMI_ENABLE_STRIPING=1
- export PAMI_IBV_ADAPTER_AFFINITY=1
- Export PAMI_IBV_DEVICE_NAME="mlx5_0:1,mlx5_3:1"
  - Small latency bound messages will only use one port



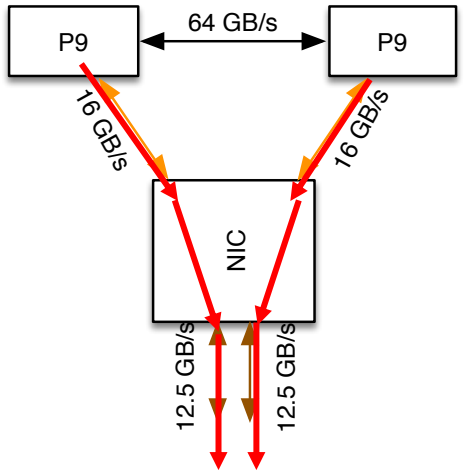
Example Configurations – 6 Ranks Full Bandwidth

- export PAMI_IBV_ENABLE_DCT=1
- export PAMI_ENABLE_STRIPING=1
- export PAMI_IBV_ADAPTER_AFFINITY=1
- Export PAMI_IBV_DEVICE_NAME=**“mlx5_0:1,mlx5_3:1”**
- Export PAMI_IBV_DEVICE_NAME=**“mlx5_3:1,mlx5_0:1”**
  - **Tests using all 4 ports resulted in 33% lower bandwidth**



Example Configurations – 6 Rank Latency Bound

- export PAMI_IBV_ENABLE_DCT=(0 Smaller job 1 Larger job)
- export PAMI_ENABLE_STRIPING=0
- export PAMI_IBV_ADAPTER_AFFINITY=1
- Export PAMI_IBV_DEVICE_NAME="mlx5_0:1" (Port per socket)
- Export PAMI_IBV_DEVICE_NAME_1="mlx5_3:1"



Recommended for most jobs (dual socket)

- PAMI_IBV_ENABLE_STRIPING=1
- Now
  - PAMI_IBV_DEVICE_NAME="mlx5_0:1,mlx5_3:1"
  - PAMI_IBV_DEVICE_NAME_1="mlx5_3:1,mlx5_0:1"
- After March if significant cross bus GPU traffic
  - PAMI_IBV_DEVICE_NAME="mlx5_0:1,mlx5_1:1"
  - PAMI_IBV_DEVICE_NAME_1="mlx5_3:1,mlx5_2:1"
  - Current this will break until SMPI supports relaxed ordering
- Testing Shows
  - More resilient to adversarial congestion
  - Very small latency cost to crossing the X-Bus (100ns)
  - Increased bandwidth

SHARP

- Scalable Hierarchical Aggregation (and) Reduction Protocol
  - Means: Our network builds fancy trees in switches to accelerate some collective operations
  - Supported Collectives (Small <= 2048)
    - Barrier
    - Broadcast
    - Reduce
    - Allreduce

SHARP Performance Measurements

- Barrier
  - 6us@512 nodes vs 21-23 for software
- Allreduce
  - 18us@2048 nodes vs 85 - 139

Things to know

- It's a shared resource
  - You may request it and not get it, we're imposing allocation policies that favor jobs > 1% of the machine.
  - If you use a lot (a lot) of sub-communicators
    - It creates an OST tree for every communicator group
- Small collectives
- Bitwise reproducibility
  - OST locations are dynamic and change

How to use it

- ENABLE_SHARP="-E HCOLL_ENABLE_SHARP=2 -E HCOLL_SHARP_NP=2 -E SHARP_COLL_LOG_LEVEL=3 -E HCOLL_BCOL_P2P_ALLREDUCE_SHARP_MAX=2048 -E SHARP_COLL_JOB_QUOTA_OSTS=64 -E SHARP_COLL_POLL_BATCH=1 -E SHARP_COLL_SHARP_ENABLE_MCAST_TARGET=0 -E SHARP_COLL_ENABLE_MCAST_TARGET=0 -E SHARP_COLL_JOB_QUOTA_PAYLOAD_PER_OST=256”
- ENABLE_HCOLL="-mca coll_hcoll_enable 1 -mca coll_hcoll_np 0 -mca coll ^basic -mca coll ^ibm -HCOLL -FCA”

- jsrun -n … -r … $ENABLE_SHARP --smpiargs="$ENABLE_HCOLL”

Required Flags

- HCOLL_ENABLE_SHARP=
  - 1 (Probe and use it) Falls back to HCOLL if unsuccessful
  - 2 (Force use it) Falls back to application failure if unsuccessful
  - 3 & 4 Various nuances on 2
- -mca coll_hcoll_enable 1 //SHARP is built into HCOLL
- -mca coll ^basic   //Disable both spectrum collectives
- -mca coll ^ibm
- -HCOLL //Force HCOLL
- -FCA //Force Fabric Collective Accelerator

Some other advanced capabilities (Coming soon)

- GPU Direct RDMA Async
  - Initiate HCA transfer from GPU (Requires code changes)
- Hardware tag matching
  - Reduce overhead of matching rendezvous messages
- Tunneled Atomic Operations
  - RDMA Write payload indicates which host side atomic operation to perform

Conclusion

- There are lots of options to tune in the Summit network
  - Port mapping can play a large roll in network performance
  - Always use Adaptive routing
  - Stripe for bandwidth
  - SHARP configuration can get complicated but even basic tuning should see significant improvements to collective performance at large scale
- Questions?
  - Feel free to email me with questions:
    - zimmercj@ornl.gov