Improving Data Movement Performance for Sparse Data Patterns on the Blue Gene/Q Supercomputer

Huy Bui, Jason Leigh

Electronic Visualization Laboratory, University of Illinois at Chicago

Eun-Sung Jung, Venkatram Vishwanath, Michael E. Papka

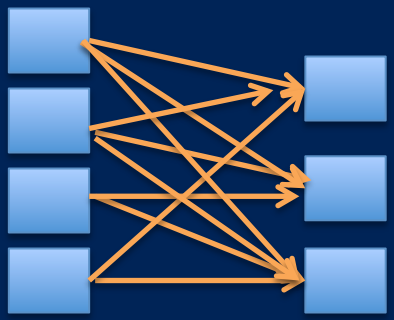
Argonne National Laboratory

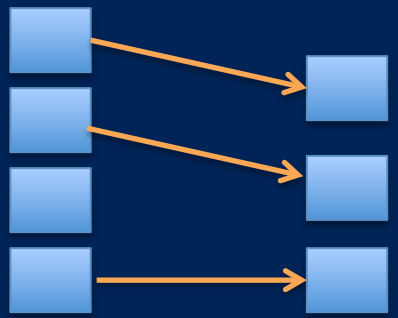
Sept 12th, 2014

Minneapolis, Minnesota, USA

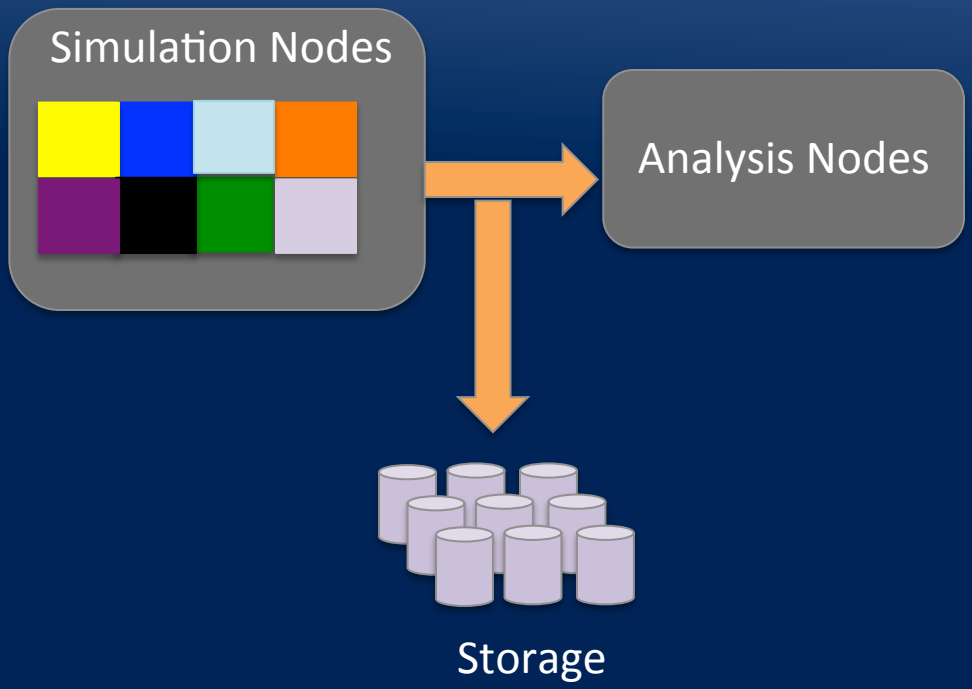
Sparse data movement

- Dense communication:
  - Lot of nodes/links involve into communication.
- Sparse communication:
  - Small number of nodes/links involve in communication: 10%-30% of nodes.

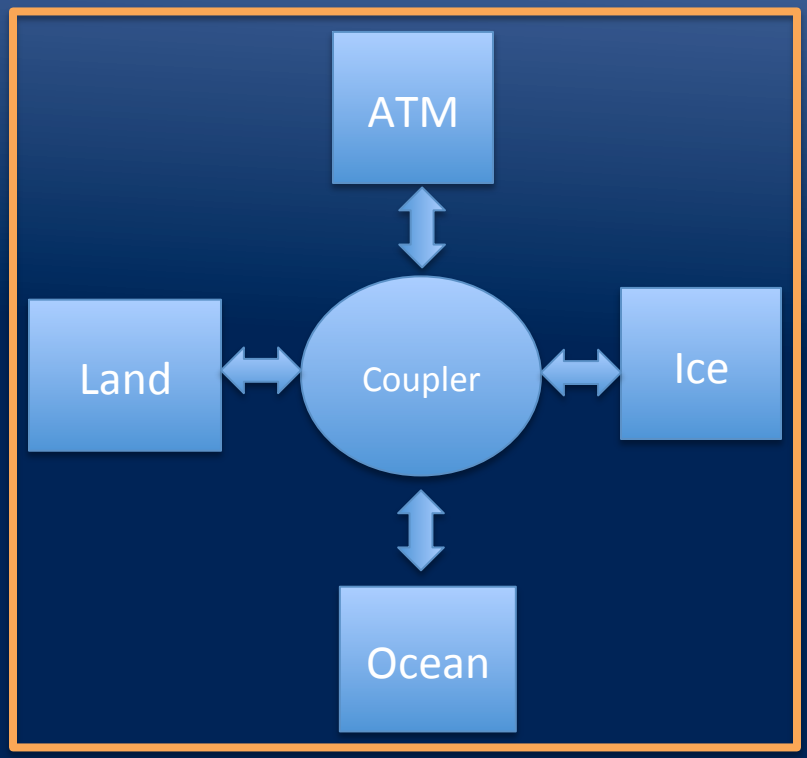




Sparse Data Movement Patterns



in situ Analysis

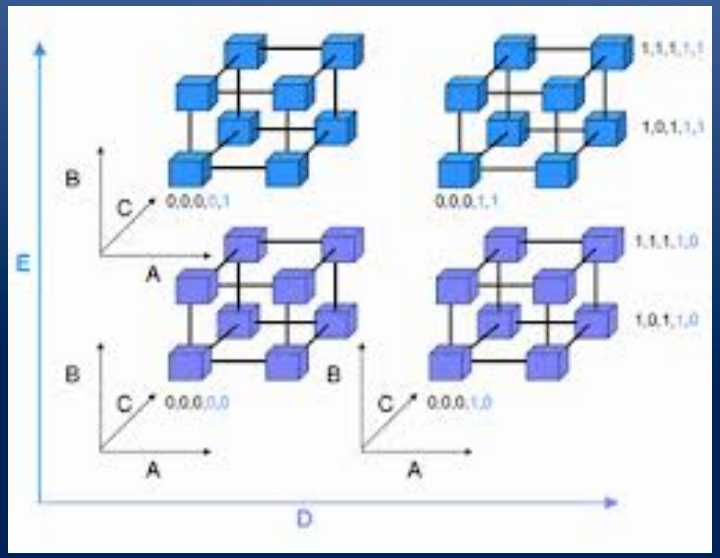


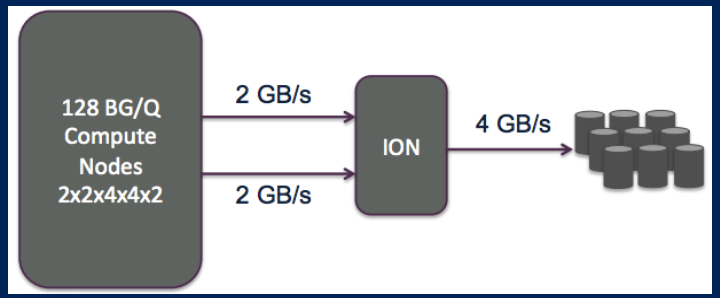
Sparse communication

Data Movement in Multi-Physics Model Coupling

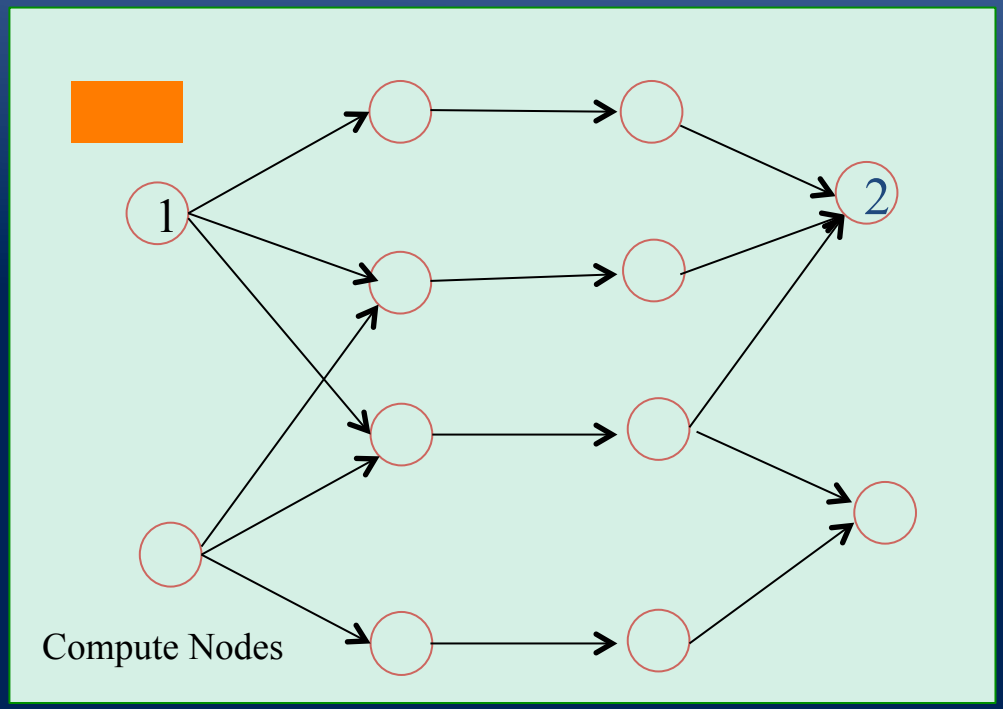
The Blue Gene/Q supercomputer

- Interconnection network:
  - 5D interconnection network: A, B, C, D, E directions.
  - Each node has at most 9 neighbors (2 in each A, B, C, D direction, 1 in E direction).
  - 10 links per node with 2GB/s per link (~1.7GB/s is effective for users).
- I/O
  - One I/O node per 128 compute nodes.
  - Via 2 bridge nodes (among 128 nodes): 2 x 2GB/s link.





Inefficiencies of sparse data movement in the Blue Gene/Q



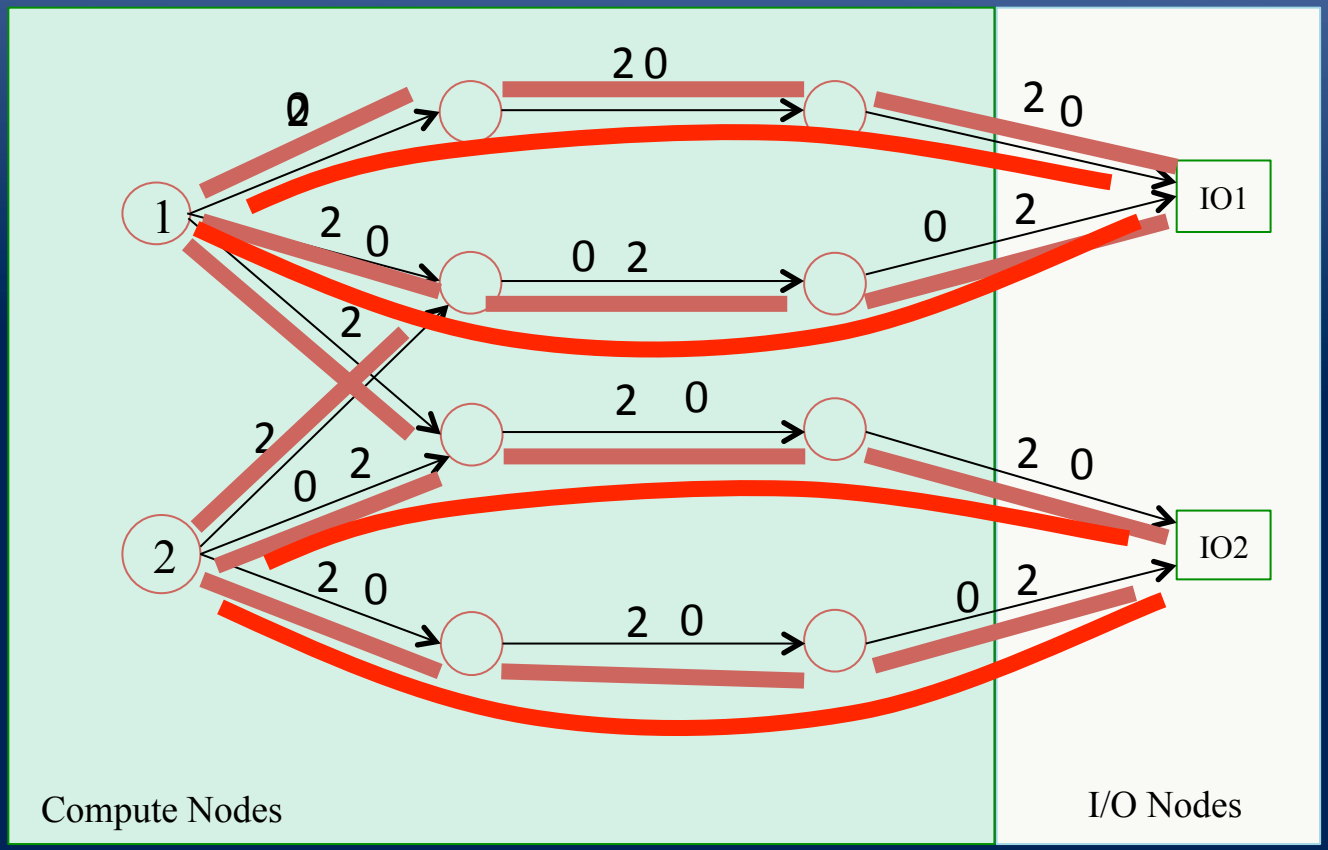
Communication between compute nodes.

- Single path is used while multiple paths are available.
- Similar inefficiencies are also encountered when we move data to the BG/Q IO nodes during I/O operations.

Improving sparse data movement

- Model the interconnection network as graph:
  - Each compute node as a vertex.
  - Each physical link as an edge.
  - The problem is to find multiple paths to transfer data from source(s) to destination(s).
- Exploit multiple paths from a source node to a destination node using Ford-Fulkerson algorithm.
- Introduce Proxies (intermediate nodes) for data movement from sources to destination that do not follow default system routing policies.

Finding Multiple Paths

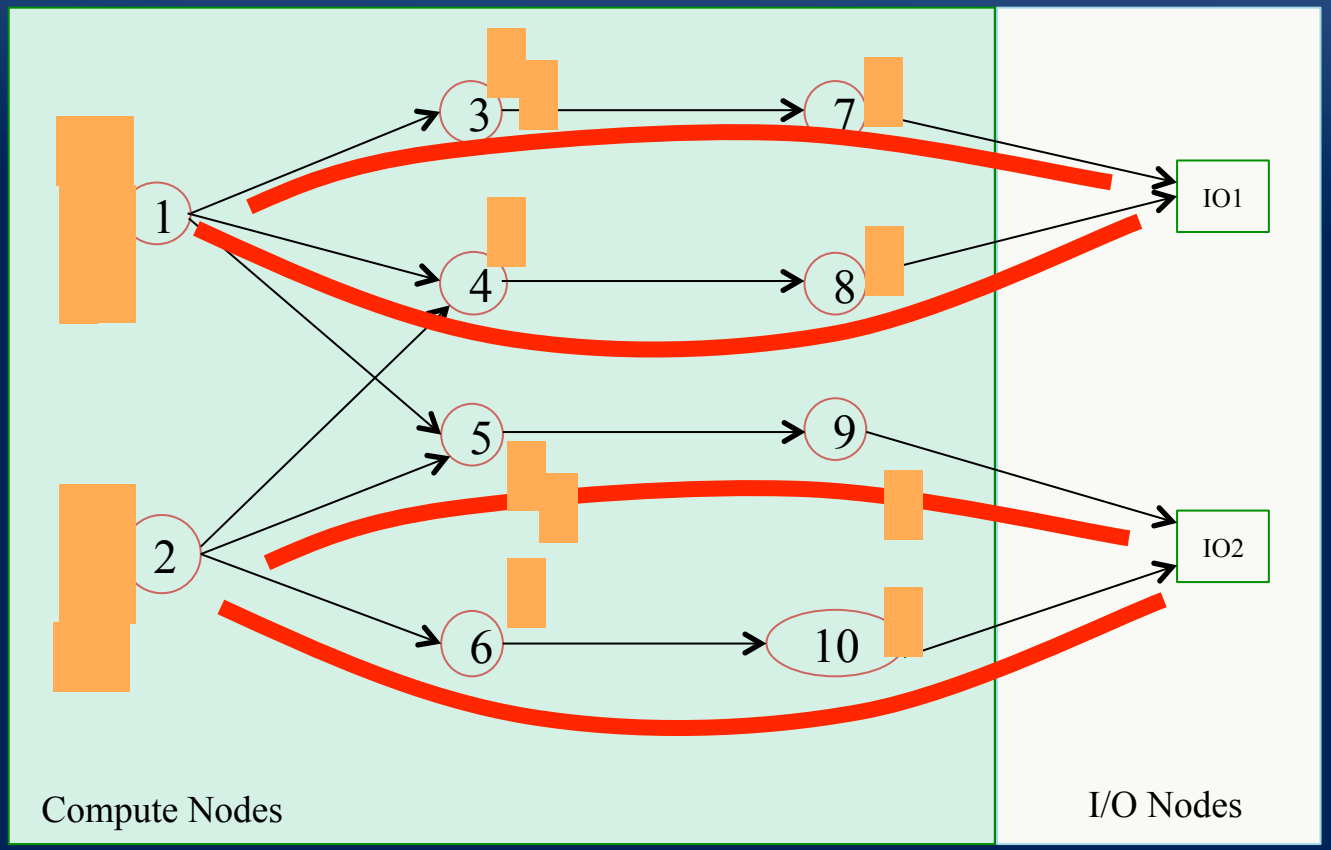


Using Ford-Fulkerson to find multiple paths using maximum flow between multiple sources and multiple destinations heuristically.

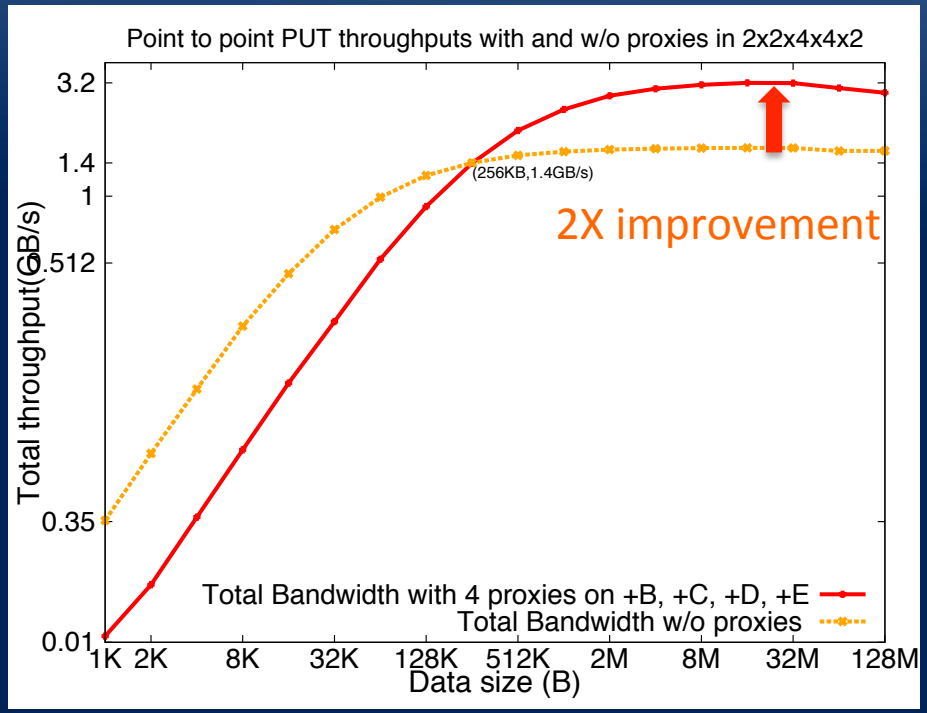
Proxies for Data Movement

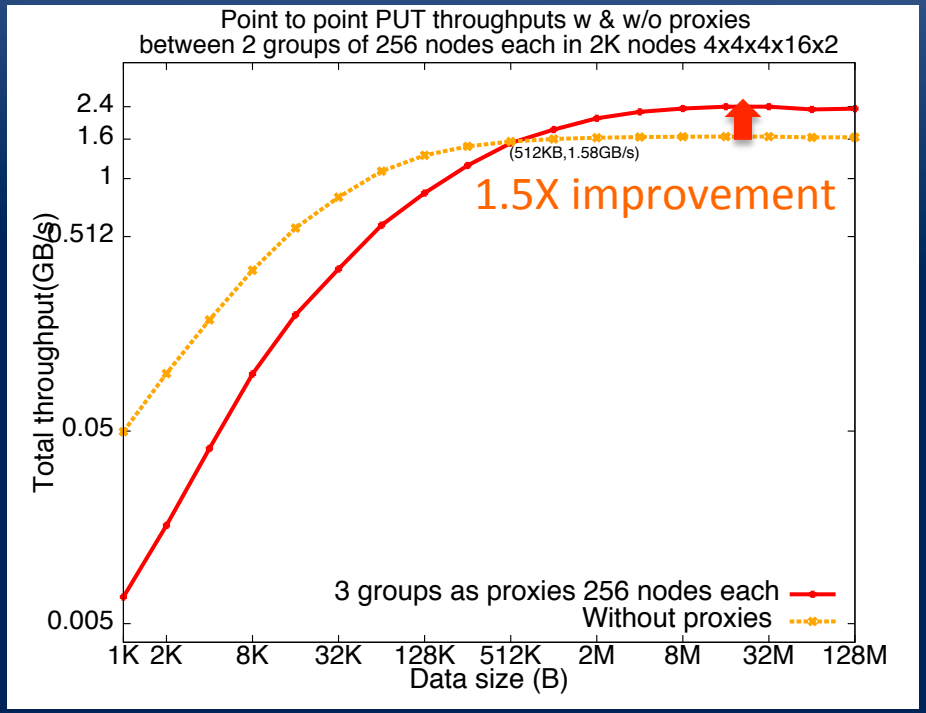
- To leverage multiple paths, we introduce proxies for moving data from source to destination.
- The data now needs to traverse the multiple intermediate proxies and incurs an overhead of the additional data movement due to the proxies.
- This overhead can be reduced :
  - by using data pipeline techniques to overlap communication.
  - By careful selection of proxy nodes to exploit the default system routing.

Multi-path data movement between groups of nodes



Efficacy on data movement between groups of nodes

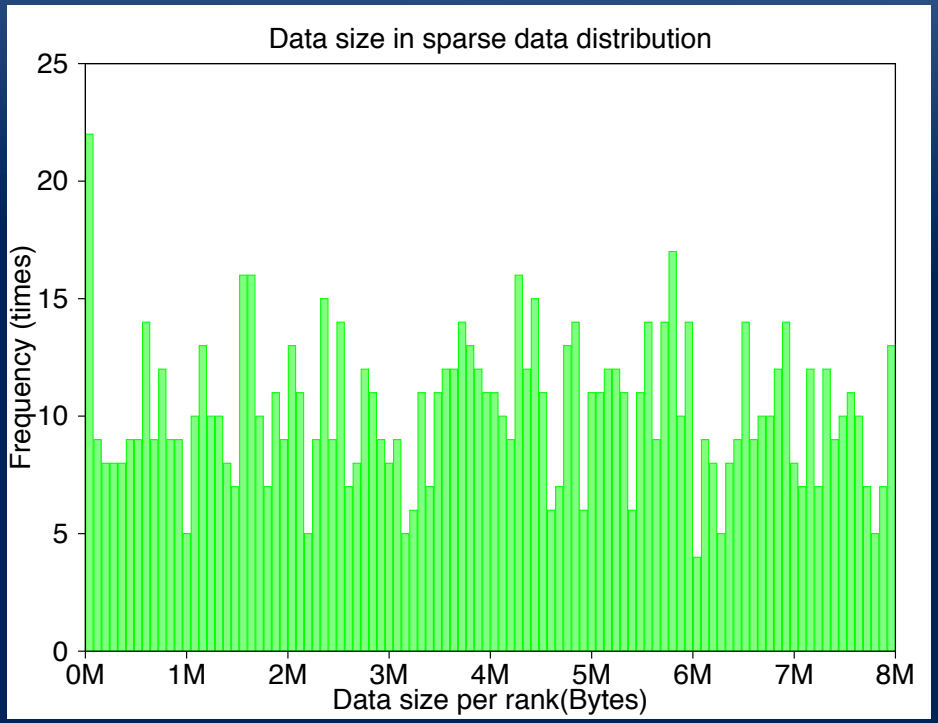


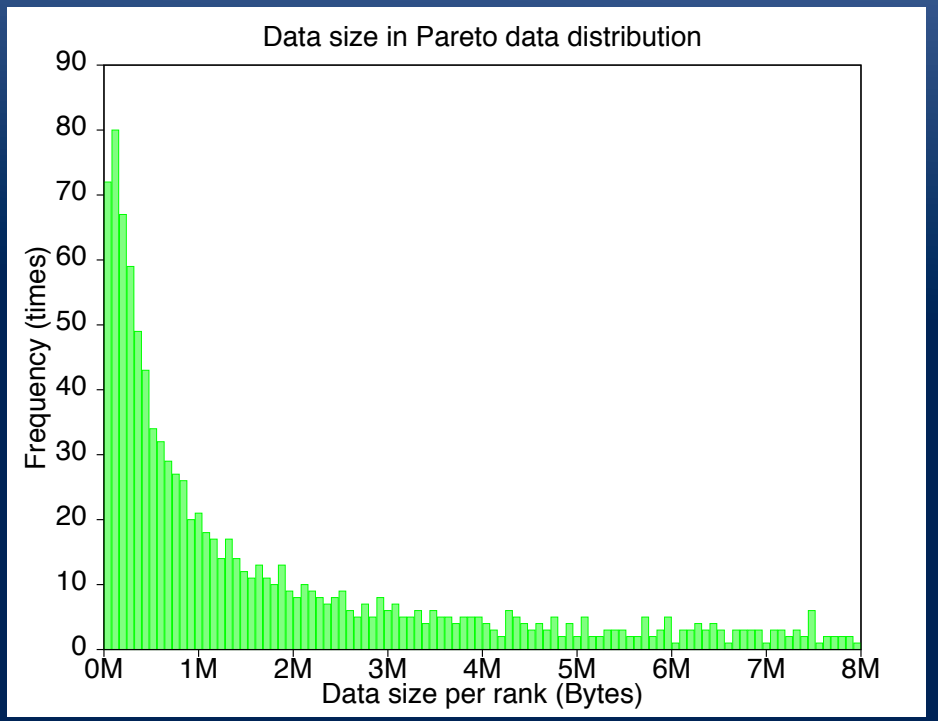


Point to point throughput 1 pair of nodes in 256 nodes partition with 4 paths.

Point to point between 256 pairs of nodes in 2K nodes partition via 3 paths/pair.

Sparse I/O Data Patterns

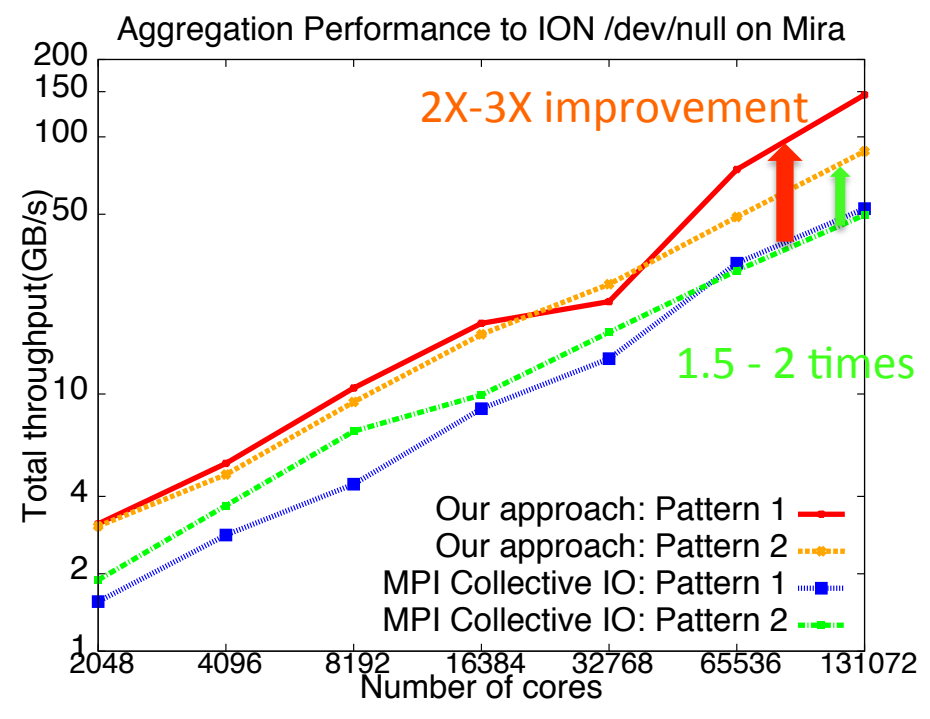




Uniform data size distribution: Data size is varied but uniformly distributed among physical nodes.

Pareto data size distribution: Most of nodes have no data or very small size of data.

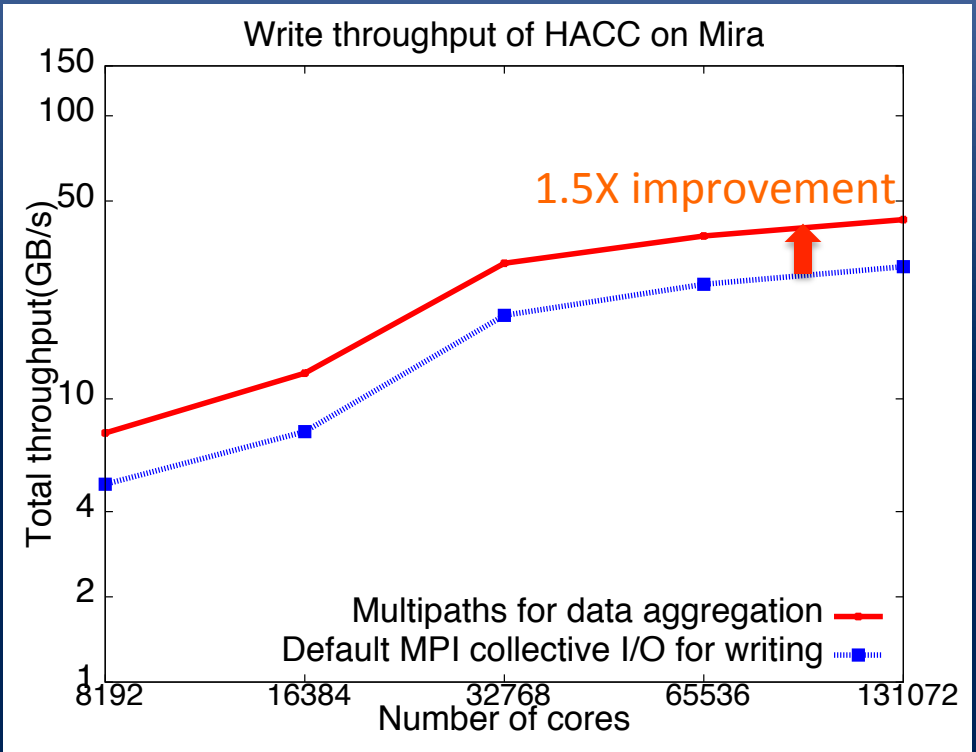
Weak Scaling Performance to 128K cores



Aggregation throughputs on Mira

- Generated data follow the 2 patterns (uniform and Pareto).
  - Pattern 1: 8 GB to 274 GB.
  - Pattern 2: 3.4 GB to 119 GB.
- Randomly place data on compute nodes.

Performance with HACC I/O



Write throughput of HACC application to I/O nodes /dev/null.

- HACC – Hybrid /Hardware Accelerated Cosmology Code: data intensive applications.
- Choose to write 10% of data to I/O from 2GB to 85 GB of data.
- Multiple paths can improve significant performance for sparse I/O.

Conclusions and Future work

- Conclusions:
  - Topology-aware data movement is important for sparse data patterns.
  - Multiple paths data movement can improve performance significantly for sparse data patterns.
- Works to be incorporated into Journal version:
  - Pipeline technique to reduce overheads and memory used due to copy and forward.
  - Using PAMI for better performance with small messages.

Thank you!