Topology-aware data aggregation for parallel I/O on Blue Gene/Q

François Tessier, Preeti Malakar, Venkatram Vishwanath, Emmanuel Jeannot, Florin Isaila
Email: `ftessier@anl.gov` - Argonne Leadership Computing Facility

Argonne NATIONAL LABORATORY

Reading and writing data efficiently from storage sytems is critical for high performance data-centric applications. These I/O systems are being increasingly characterized by complex topologies and deeper memory hierarchies. Effective parallel I/O solutions are needed to scale applications on current and future supercomputers.

General Overview

- The gap between the computational capacity and I/O performance of supercomputers is growing

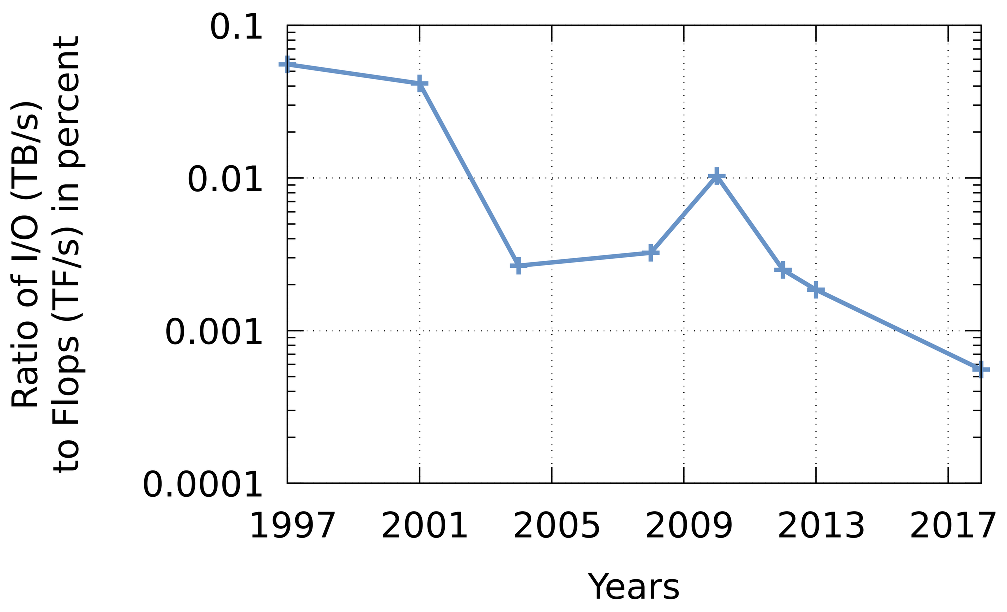

Figure 1: The performance ratio between the I/O bandwidth and the computing capability of the #1 Top 500 for the past 20 years. Computing capability has grown at a faster rate than the I/O performance of supercomputers.

- Higher resolution and higher fidelity scientific simulations have high I/O requirements

Scientific domain	Simulation	Data size
Cosmology	Q Continuum	**2 PB / simulation**
High-Energy Physics	Higgs Boson	**10 PB / year**
Climate / Weather	Hurricane	**240 TB / simulation**

Performance of parallel I/O on current supercomputers

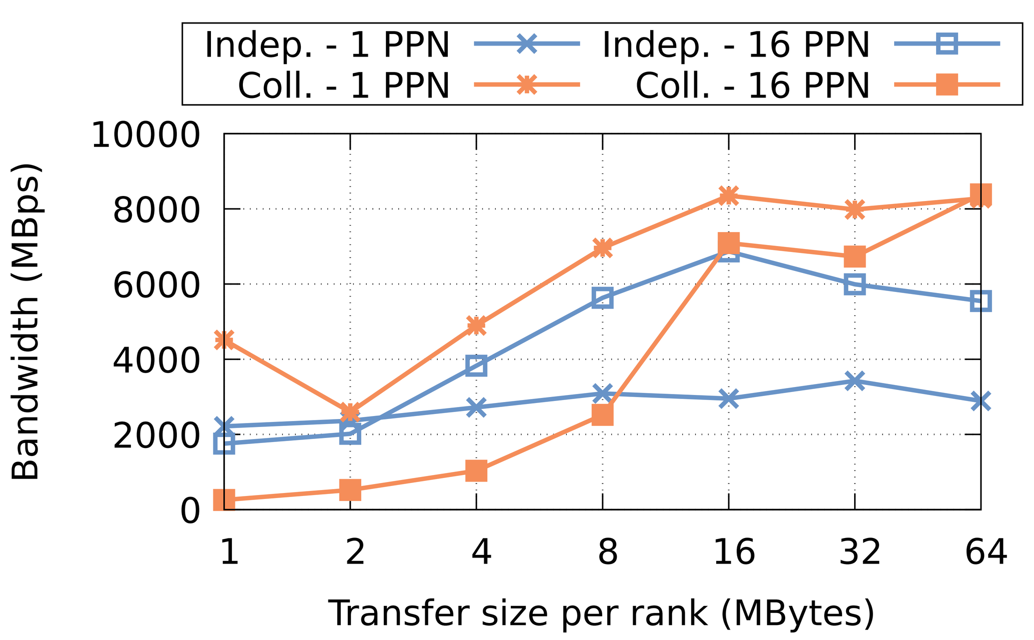

Figure 2: Write bandwidth for a single shared file using the IOR benchmark on 512 nodes of the Mira BG/Q system with 64 MB/rank as we vary the transfer size.

- Current mechanism work best for large messages and fewer ranks per node.
- As we scale to future systems with larger core counts per node and lower memory per core, effective parallel I/O algorithms are needed.

Data Aggregation

- Collect data from processes to write larger messages
- Reduce network contention
- Increase I/O bandwidth

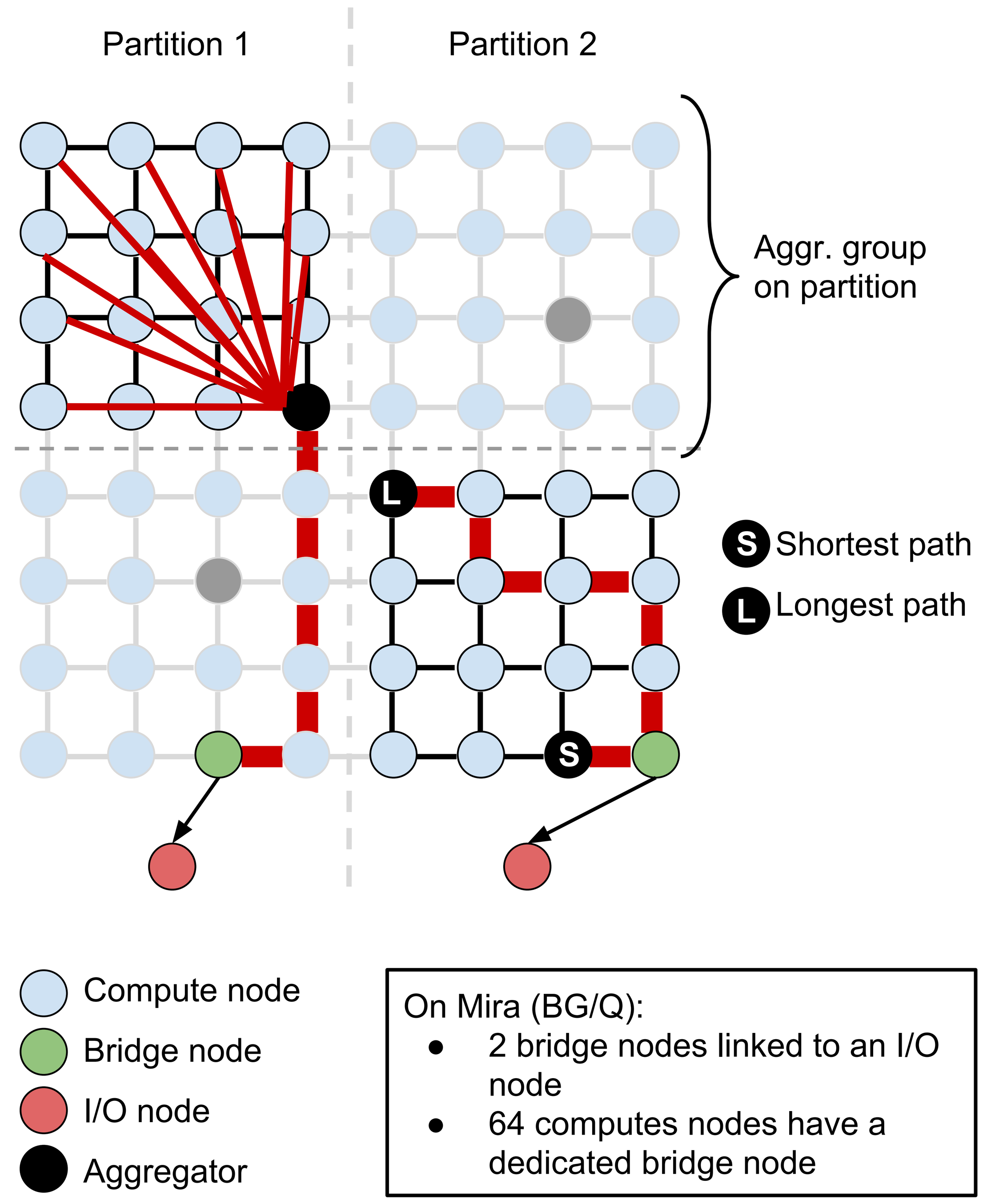

Figure 3: Data aggregation for I/O

Challenges at scale

- Complex interconnect on supercomputers (5D Torus, Dragonfly) and complex/deep memory hierarchy (MCDRAM, DRAM, NVRAM)
  - Where to place aggregators in the topology?
- What is an efficient number of aggregators to manage data?

Analytical Model

Number of aggregators

Given:

- BS: The block size of the target file system
- D_{tot}: The total amount of data to be written
- Mem: A available memory for an aggregator
- $\#Aggr$: The number of aggregators to select

- Find a number of aggregators such that:
  - The aggregator writes more data than the block size to mitigate file system overheads such as locking
  - Current limitation: $\#Aggr$ is a power of 2

$$\#Aggr = \lceil \log_2 \left(\frac{D_{tot}}{Mem}\right) \rceil^2, \frac{D_{tot}}{Mem} > BS$$

Topology-aware placement of aggregators

Given:

- V_C: The set of compute nodes performing aggregation
- $\omega(u,v)$: The data size exchanged between nodes u and v
- $A \in V_C$: An aggregator chosen among compute nodes
- l: The interconnect latency
- B: The bandwidth between two compute nodes
- $d(u,v)$: The number of hops between nodes u and v
- IO: The I/O node

- Sending data from compute nodes to the aggregator

$$C_1 = max\left(l \times d(i,A) + \frac{\omega(i,A)}{B_{i \to A}}\right), i \in V_C, i \neq A$$

- Sending data from the aggregator to the I/O node

$$C_2 = l \times d(A, IO) + \frac{\omega(A, IO)}{B_{A \to IO}}$$

- Our objective function consists of minimizing the sum

$$TopoAware(A) = min\,(C_1 + C_2)$$

Limits

- One aggregation round. What if $D_{tot} > Mem$?
- Impact of the other aggregators in network contention (routing)
- Regular distribution of aggregators in partition

Results

Experiments

- HACC-IO: I/O part of Hardware Accelerated Cosmology Code
- Architecture
  - 4096 nodes on Mira with 16 PowerPC A2 cores, 1600 MHz (65536 cores)
  - 5D Torus network, 1.8 GBps per link
  - 1 GB Memory per core
  - GCC v4.4.7, MPICH2 v1.5
- Compared to two greedy strategies and MPI collective I/O

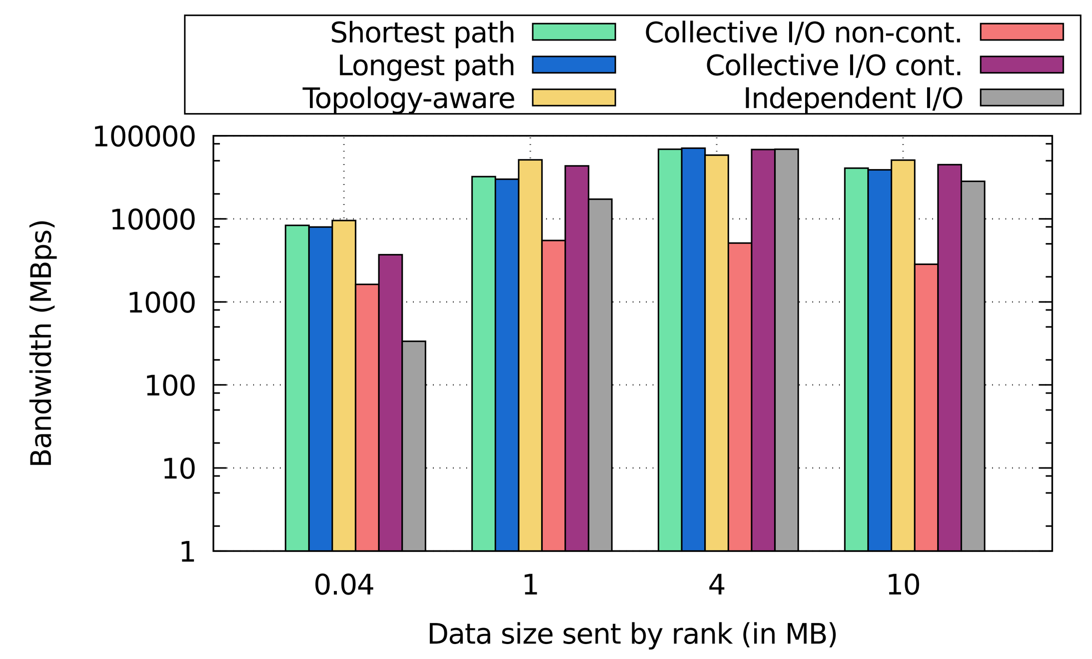

Figure 4: Write bandwidth for each aggregator mapping strategy according to the data size

Observations

- Improved performance in case of collective writes for non-contiguous data over the default collective I/O
- Up to 180% performance improvment for small messages compared to collective write of contiguous data
- Topology-aware approach leads to better performance than greedy strategies except for 4MB message size

Future work

- Scale to a larger core counts
- Expand to include more varieties of data patterns
- Generalize to incorporate other supercomputing architectures