Short Talk: System abstractions to facilitate data movement in supercomputers with deep memory and interconnect hierarchy

François Tessier, Venkatram Vishwanath

Argonne National Laboratory, USA

July 19, 2017

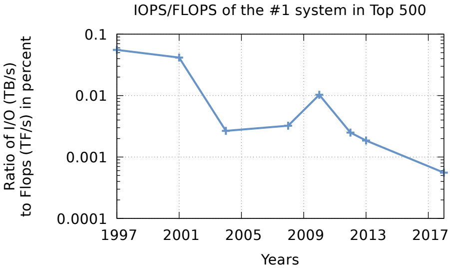
Data Movement at Scale

- Computational science simulation such as climate, heart and brain modelling or cosmology have large I/O needs
  - Typically around 10% to 20% of the wall time is spent in I/O

Table: Example of I/O from large simulations

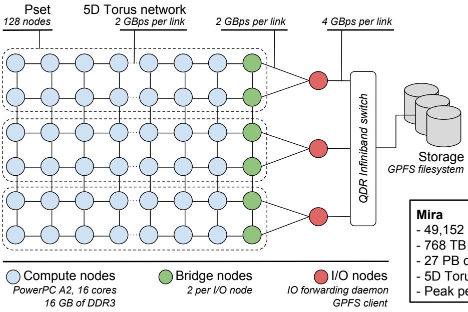
Scientific domain	**Simulation**	**Data size**
Cosmology	Q Continuum	**2 PB / simulation**
High-Energy Physics	Higgs Boson	**10 PB / year**
Climate / Weather	Hurricane	**240 TB / simulation**

- Increasing disparity between computing power and I/O performance in the largest supercomputers



Complex Interconnect Hierarchies

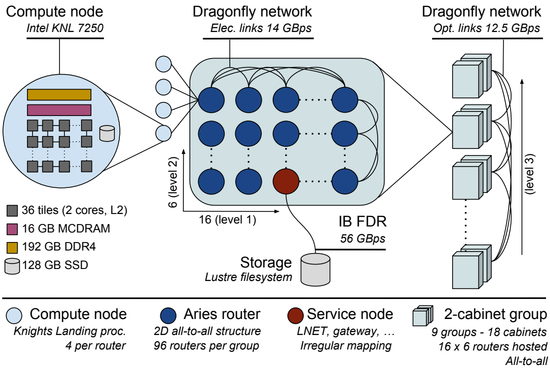
- On BG/Q, data movement needs to fully exploit the 5D-Torus topology for improved performance
- Additionally, we need to exploit the placement of the I/O nodes for performance
- Cray supercomputers have similar challenges with dragonfly-based interconnects together with placement of LNET nodes for I/O



Mira
- 49,152 nodes / 786,432 cores
- 768 TB of memory
- 27 PB of storage, 330 GB/s (GPFS)
- 5D Torus network
- Peak performance: 10 PetaFLOPS

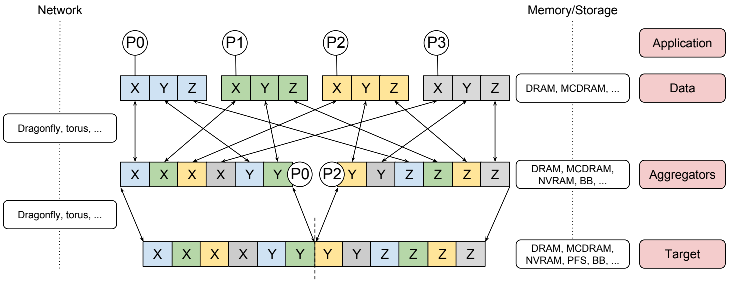
Deep Memory Hierarchies and Filesystem characteristics

- We need to exploit the deep memory hierarchy tiers for improved performance
  - This includes effective ways to **seamlessly** use HBM, DRAM, NVRAM, BurstBuffers, etc.
- We need to leverage filesystem specific features such as OSTs and striping in Lustre, among others.



TAPIOCA - Ongoing research

- Library based on the two-phase I/O scheme for topology-aware data aggregation at scale on IBM BG/Q with GPFS and Cray XC40 with Lustre (Cluster'17, JLESC Collaboration with Emmanuel Jeannot@Inria)
  - Topology-aware aggregator placement
  - Pipelining (RMA, non-blocking calls)
  - Interconnect architecture abstraction
- Move toward a **generic data movement library for data-intensive applications** exploiting deep memory/storage hierarchies as well as interconnect to facilitate I/O, in-transit analysis, data transformation, data/code coupling, workflows, ...



What is the right level of abstraction?

A specific abstraction for every system including the architecture, filesystems, capturing every phase of deployment, relevant software versions, etc.

A generalized abstraction that maps to current and expected future deep memory hierarchies and interconnects (including performance, contention, etc.)

The abstractions and tradeoffs for performant and portable data movement

Abstractions for Interconnect Topology

- Topology characteristics include:
  - Spatial coordinates
  - Distance between nodes: number of hops, routing policy
  - I/O nodes location, depending on the filesystem (bridge nodes, LNET, ...)
  - Network performance: latency, bandwidth
- Need to model some unknowns and uncertainties such as routing, contention

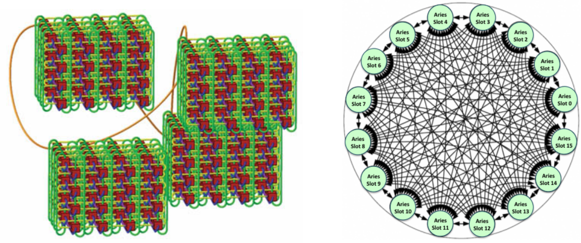
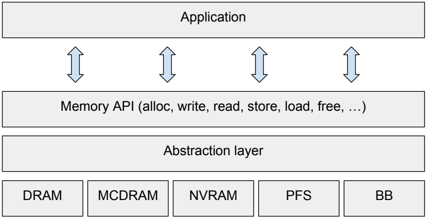


Figure: 5D-Torus on BG/Q and intra-chassis Dragonfly Network on Cray XC30 (Credit: LLNL / LBNL)

Abstractions for Memory and Storage

- Topology characteristics including spatial location, capacity and distance
- Performance characteristics including bandwidth, latency and support for concurrency
- Access characteristics such as byte-based vs block based
- Persistency



Need to account for application needs in I/O, in-situ vizualisation, in-situ analysis, data transformation, workflows, etc. and map these onto the underlying abstractions for improved performance.

Conclusion

Thank you for your attention!

ftessier@anl.gov