
Thoughts on Future Directions in Exascale

William Gropp
<http://wgropp.cs.illinois.edu>

What am I Counting on for Exascale Architectures?

- I'd like a single, fast processor just like everyone else. RSFQ anyone?
- Expect to have a heterogeneous system with
 - Specialized processors for vectors and streams
 - Bandwidth-optimized memory
 - General purpose processors for “control-flow” critical code
 - Nonblocking, high-bandwidth interconnects
- Still looking for a solution for applications requiring strong scaling and those that are latency dominated
 - Some of this can be fixed at the software or algorithm level
 - See RSFQ above
- Expect more specialized compute elements in the future
 - FPGA ecosystem continues to improve
 - End of Dennard scaling has created an opportunity
 - Still requires sufficiently important operations/applications to justify all costs

Are HPC and Big Data Converging?

- Yes and no
- Yes – Techniques from HPC are (finally!) being adopted in Big Data
 - E.g., Baidu discovers scalable Allreduce (and credits HPC)
 - <http://research.baidu.com/bringing-hpc-techniques-deep-learning/>
 - IBM uses MPI in its distributed DNN library PowerAI DDL
 - <https://arxiv.org/pdf/1708.02188.pdf>
- No – HPC I/O is a poor match for big data applications (and for scientific applications)
 - More on the next slide
 - Even with that fixed (techniques from big data (finally!) being adopted by HPC), different usage patterns including lifetime of data
 - Likely to have convergence in technologies but system configurations likely to be different
 - Expect different systems, but designed to be integrated into single workflows

No Science Application Code Needs POSIX I/O

- Many are single reader or single writer
 - Eventual consistency is fine
- Some are disjoint reader or writer
 - Eventual consistency is fine, but must handle non-block-aligned writes
- Some applications use the file system as a simple data base
 - Use a data base instead – we know how to make these fast and reliable
- Some applications use the file system to implement interprocess mutex
 - Use a mutex service – even MPI point-to-point
- A few use the file system as a bulletin board
 - May be better off using RDMA
 - Only need release or eventual consistency
- Correct Fortran codes do not require POSIX
 - Standard **requires** unique open, enabling correct and aggressive client and/or server-side caching, e.g.
- MPI-IO would be better off without POSIX
 - Does not and never has required POSIX

Is AI a new “Killer App” for HPC?

- Lets be more precise about our terms
 - What we really have is *Imitation intelligence*
 - The appearance of intelligence without anything behind it
 - Still incredibly powerful and useful, but ...
 - Not *Artificial intelligence*
 - Intelligence achieved through artificial means
- This is important for this question because of the lack of adaptability of current “AI” solutions
 - Training required for each “behavior” (one reason this is II, not AI)
 - Current methods require large amounts of data and compute to train; application of the trained system is not (relatively speaking) computationally intensive
- Thus HPC, defined as systems that can deliver high performance, and that are operated by, and supported by, experts that understand performance issues and can diagnose and repair performance problems, is important and could be a “killer technology” for Imitation Intelligence
- But we will need the correct software and specialist ecosystem for this to be true