Planes, Trains, and Automobiles: Making Your Data Available in Time for Discovery

Quincey Koziol (koziol@lbl.gov) and Suren Byna (sbyna@lbl.gov)

Topic: Architectures, Emerging technologies

Challenge

Data movement in high performance computing (HPC) continues to be a challenging issue today and is on track to become even more complex in future systems. HPC systems are becoming increasingly heterogeneous in their compute elements with CPUs, GPUs, and special-purpose FPGAs. As a result, the memory locations that buffer for data for processing are intrinsically heterogeneous. In addition, storage devices themselves are becoming heterogeneous, with new non-volatile and memory-class memories, although traditional parallel file systems are here to stay as capacity storage. Software that manages each of these memories and storage devices are diverse, i.e., memories are managed by processor hardware and operating systems, whereas storage devices are managed by parallel file systems.

Applications are not currently designed with this complex memory/storage hierarchy in mind and typically offload data to persistent storage in a generic manner, very similar to their traditional interactions with POSIX-based file systems. Needless to say, the lack of information from the application about how important and persistent its data is presents challenges to storage system stack. High- and low-level I/O middleware, such as HDF5 and MPI-I/O, attempt to gather enough information to perform I/O operations efficiently, but are not able efficiently leverage the storage system without more information about the data from the application, as well as more information about the storage stack from the system. All these layers, applications, memory management, I/O middleware, parallel file system, and the HPC system itself, must collaboratively build better models of where, when, and how to move data to satisfy the needs of application users. A co-design effort is essential to utilize the capabilities of heterogeneous memory and storage devices to their full potential.

Opportunity

Future application co-design must focus on data movement as much as compute efficiency. Energy costs for data movement are a growing factor in system design, and are projected to be the dominant factor in the future as benefits from Moore's Law decrease density increases per silicon die and future system performance gains will likely be made through adding more cores, with corresponding increases in interconnections and therefore data movement [1, 2].

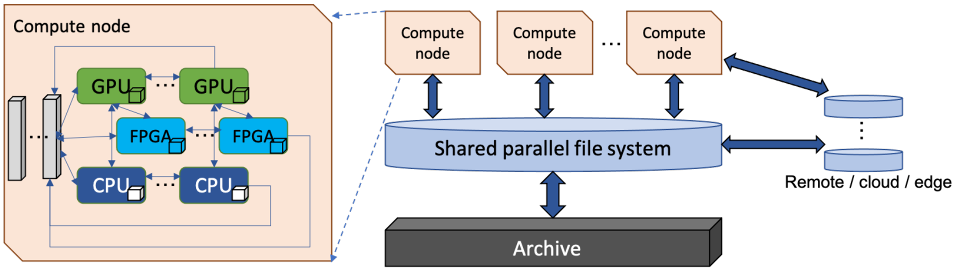
We propose three areas of focus for data movement co-design:

1. Application Guidance and Input - Applications, whether simulation, experimental / observational data focused, or AI-based, must add more information to their data movement requests. No longer can an application just make a POSIX 'write' call to send a buffer to persistent storage. The application should describe the importance and persistence of the buffer to the data movement middleware and participate in negotiating shared ownership of system resources like compute-local memory, system interconnect, node-local storage, and in-network compute with the storage middleware. Interfaces are needed for applications to express the data movement irrespective of heterogeneous end-points.

2. Data Movement Middleware - High- and low-level data movement middleware, similar to today's I/O middleware such as HDF5 and MPI-I/O, must expose a more flexible set of options for applications to choose from. These could include synchronous vs. asynchronous I/O, borrowed, shared, or transferred ownership of memory buffers, mechanisms for recognizing data importance and persistence priorities, etc. Storage middleware must also query a system's configuration: What layers are there in the memory and storage hierarchy? How much space is available in them? How are those layers connected and how fast are those connections? Only

when these questions can be answered will the storage middleware be able to optimize data movement for the application.

3. HPC Runtime Systems - Runtime systems that schedule data movement, manage available space, and place data closer to analysis capabilities can achieve high efficiency, but the systems' memory/storage hierarchy must also become "introspectable" and "programmable". Data movement middleware must be able to retrieve the system's configuration, both statically (to know maximum capabilities) and dynamically (to adjust data movement according to current system load). In addition, flexible and programmable capabilities to transfer data between all layers of the system's memory/storage hierarchy must be exposed for the storage middleware to leverage as needed.



As the diagram above shows, future HPC systems' memory/storage hierarchy will have many "moving pieces" and will need careful optimization in a concerted and organized way. Application developers alone will not wish to invest the required effort involved to wring the best performance possible from complex storage systems as their changes will require rework when porting to a new system. System vendors also don't have a vested interest in building portable data movement middleware that might benefit competitors in the field.

Instead, this era will see the rise of data movement middleware that provides a rich and flexible interface for application developers to express their data movement desires and will also query the system's memory/storage hierarchy for all the information needed to fully carry out those requests. Indeed, the "data movement middleware" that is required to help application teams and system vendors extract the maximum benefit from the underlying system will no longer be focused on merely accessing "stored" data - it will instead be tasked with the entire, complex, data movement process, from RAM to tape and the cloud, and all steps along the way.

Timeliness: These data movement codesign efforts will turn the typical compute-focused codesign effort on its head - compute may play a secondary role to the overall optimization of getting data back to science teams in an efficient, timely, and easy-to-manage way. Only now, in the last days of Moore's Law, are the costs and importance of optimizing for data movement being revealed and coming to the forefront of codesign.

References

1. P. Kogge and J. Shalf, "Exascale Computing Trends: Adjusting to the New Normal for Computer Architecture," in Computing in Science & Engineering, vol. 15, no. 6, pp. 16-26, Nov.-Dec. 2013, DOI: 10.1109/MCSE.2013.95.
2. J.S. Vetter, et al., "Extreme Heterogeneity 2018: DOE ASCR Basic Research Needs Workshop on Extreme Heterogeneity", US Department of Energy, Office of Science, Advanced Scientific Computing Research, 2018, DOI:10.2172/1473756.