ExaIO

In pursuit of more accurate modeling of real-world systems, scientific applications at exascale will generate and analyze massive amounts of data. A critical requirement of these applications to complete their science mission is the capability to access and manage these data efficiently on exascale systems. Parallel I/O, the key technology behind moving data between compute nodes and storage, faces monumental challenges from new application workflows as well as the memory, interconnect, and storage architectures considered in the designs of exascale systems. The ExaIO project is delivering the HDF5 library and the UnifyFS tool to efficiently address these storage challenges.

Parallel I/O libraries of the future must be able to handle file sizes of many terabytes and I/O performance much greater than currently achievable to satisfy the storage requirement of exascale applications and enable them to achieve their science goals. As the storage hierarchy expands to include node-local persistent memory and solid-state storage as well as traditional disk and tape-based storage, data movement among these layers must become much more efficient and capable. The ExaIO project is addressing these parallel I/O performance and data management challenges by enhancing the HDF5 library and developing UnifyFS for using exascale storage devices.

The Hierarchical Data Format version 5 (HDF5) is the most popular high-level I/O library for scientific applications to write and read data files at supercomputing facilities and has been used by numerous applications. The ExaIO team is developing various HDF5 features to address efficiency and other challenges posed by data management and parallel I/O on exascale architectures. The ExaIO team is productizing HDF5 features and techniques that have been previously prototyped, exploring optimizations on upcoming architectures, and maintaining and optimizing existing HDF5 features tailored for the exascale applications. They are also adding new features including transparent data caching in the multi-level storage hierarchy, topology-aware I/O-related data movement, full single-writer and multi-reader for workflows, and asynchronous I/O.

Scientific applications need to periodically checkpoint the progress that has been made by saving the current state of the simulation so that the simulation can be restarted at a later time. This checkpoint/restart workflow has been reported to cause 75–80% of the I/O traffic on some high-performance computing systems. UnifyFS is a user-level file system highly specialized for shared file access on high-performance systems with distributed node-local storage that the ExaIO team is developing to specifically target checkpoint/restart workloads. UnifyFS transparently intercepts I/O calls, allowing integration of UnifyFS cleanly with other software including I/O and checkpoint/restart libraries. Thus, UnifyFS addresses a major usability factor of the pre-exascale and exascale systems.

PI: Suren Byna, Lawrence Berkeley National Laboratory

Collaborators: Lawrence Berkeley National Laboratory, Lawrence Livermore National Laboratory, Oak Ridge National Laboratory, Argonne National Laboratory, The HDF Group

Progress to date

- The ExaIO team has improved the HDF5 library in terms of performance and productivity. The team developed the Virtual Object Layer (VOL) feature to open up the HDF5 API and developed several optimizations to the improve performance of HDF5, including a topology-aware interface for the implementation of scalable algorithms and optimizations; the ability to stage the data in a temporary fast storage location, such as burst buffe, and move the data to the desired final destination asynchronously; and a capability that enables a single writing process to update an HDF5 file while multiple reading processes access the file in a concurrent, lock-free manner.

- The team completed a full system design for the UnifyFS tool and released version 2.0, which includes near complete removal of MPI dependence by integration of DataLib software for communication; support for Spack to improve the build experience and hide the complexity of UnifyFS' dependencies; and support for the Summit pre-exascale platform.