DOE Early Career Principal Investigator Award

Final Report – (08/2005 – 05/2006)

U.S. Department of Energy

DOE Award No. DE-FG02-05ER25687

Jun Wang

Computer Science and Engineering Department,

University of Nebraska Lincoln

1 Project Title

A Lightweight, High-performance I/O Management Package for Data-intensive Computing

2 Technical Progresses and Status of the Project Work

The goal of this project is to develop a lightweight, customized, high-performance I/O management package named *LightI/O* to extend and leverage state-of-the-art parallel file systems used by DOE. The PI has been working on the project with award no. DE-FG02-05ER25687 at University of Nebraska Lincoln (UNL) in Year 1 (from August 2005 to May 2006). Since that, the PI transferred to University of Central Florida (UCF) and has been continuing on the same project with award no. DE-FG02-07ER25747.

During the UNL project period, some technical progress has been made on the design, simulation and implementation of the lightweight high-performance local file system component. More specifically, the first year goal – developing a lightweight, segment-structured local file system (LSFS) has been partially obtained, in spite of a delay in hiring some new Ph.D. students. We finished an emulation study of LSFS for Parallel Virtual File System (PVFS2), and implemented a LSFS enhanced PVFS2 prototype system (i.e., LSFS+PVFS2 v1.0) on a small-scale PC based cluster. The PI’s research group named Computer Architecture and Storage System (CASS) work closely with the research group of two computer scientists Drs. Robert Ross and Rajeev Thakur and research staff Robert Latham at DOE Argonne National Laboratory (ANL in brief).

In more detail, we have successfully implemented the LSFS prototype system. We have incorporated the LSFS into PVFS2 by closely collaborating with Drs. Rob Ross and Rajeev Thakur and their research group at Argonne National Laboratory. We have conducted an emulation study and a separate real testing for LSFS+PVFS2 v1.0 by running several parallel I/O benchmarks on a small-scale PC cluster. During the development of LSFS+PVFS2, we have made the following accomplishments.

Online grouping algorithms: A grouping algorithm is developed based on the directed-graph theory to capture the group access locality and prefetch a large amount of data at local I/O.

Fault-tolerance: Reliability is addressed by saving the important system data structures in the non-volatile memory space. A high-reliability version of LSFS employs a two-pronged approach to deliver a fast recovery by writing metadata updates periodically to the reserved checkpoints. A lightweight checkpoint technique is used to maintain file system consistency while the roll-forward method is adopted to recover information written since the last checkpoint. The append-only write policy protects old data from being invalidated before new data reach disk.

Identifying limitations of local file I/O operations in state-of-the-art parallel file systems: Two limitations exist in current parallel file system solutions, which derive from lack of disk level considerations. First, there is a mapping gap between logical file data layout and physical disk data layout. Even a single large contiguous file I/O would be split into multiple small non-contiguous disk I/Os. Second, general-purpose native local file systems are employed to perform local file I/O while being proved ineffectively handle small file I/O.

Developing the LSFS+PVFS2 prototype system: We design and implement LSFS to boost the local file I/O performance for state-of-the-art parallel file systems. Parallel virtual file system (PVFS2) is chosen as an example for study. LSFS bridges the mapping gap by introducing a novel *compact segment I/O* technique, which facilitates the large-only raw disk I/O operations with the help of appropriate dynamic grouping algorithms. User-level hints using self-descriptive data structure like HDF5 or compiler hints can be used as static grouping schemes. LSFS also implements several novel lightweight and custom designs to realize sustained high-performance. At current stage, we finished the coding work of LSFS+PVFS2 v1.0 and validated it on a small-scale PC cluster.

Evaluation and Experimental Work: We have done an emulation based performance evaluation on LSFS+PVFS2. We first run real-world scientific applications and benchmarks such as mpiBLAST and mpi-Tile-IO on the PVFS2 server. Then, we logged all file I/O events into local file system traces. Afterwards, we set up a local file system test-bed comparing the LSFS prototype system with current Linux Ext3 file system. We replayed the collected traces, fed local file I/O events into two local file systems and measured the performance results in terms of file I/O response time and system throughput. The experimental results showed that LSFS boosted the file I/O response time by up to 45% and the PVFS2 system throughput by up to 72%. As of this writing, we are conducting real experiments with LSFS+PVFS2 v1.0 – an extended parallel file system on a small-scale PC cluster by running several benchmarks such as pio-bench and b_eff_io.