A position paper

# Memory-Storage Hierarchy

Osamu Tatebe
University of Tsukuba
tatebe@cs.tsukuba.ac.jp

The performance gap between CPU and storage is growing wider and wider. Big data applications and deep learning requires further storage performance than checkpointing in HPC applications. To fill the gap, a burst buffer has been proposed [1], and deployed at several supercomputing centers. The burst buffer exhibits the buffering to users explicitly and allows to know users the inconsistent state between the burst buffer and the parallel file system, which provides high storage performance. Currently, it is used to stage-in/out input/output files and to store temporal files during job executions. We, JCAHPC, deployed and operated the burst buffer for a year, and found several issues related to the burst buffer, including maturity of the software, fault handling, capacity management, performance improvement, and metadata performance. Most of these issues will be fixed in a couple of years.

Next step will be how to exploit node local non-volatile memory. Byte addressable non-volatile memory will help to improve transaction performance and write-ahead logging or journaling performance, which will be some evolution of the storage system. Key issue is how to fill the gap between high-bandwidth memory (HBM) and the parallel file system. Between them, there are DRAM, NV-RAM, NVMe SSD, and large capacity SSD. Current burst buffer solution is one approach, but there is a plenty of opportunity for system design research to efficiently utilize this memory-storage hierarchy.

Reference

[1] John Bent, et al., "PLFS: A Checkpoint Filesystem for Parallel Applications", Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis, 2009