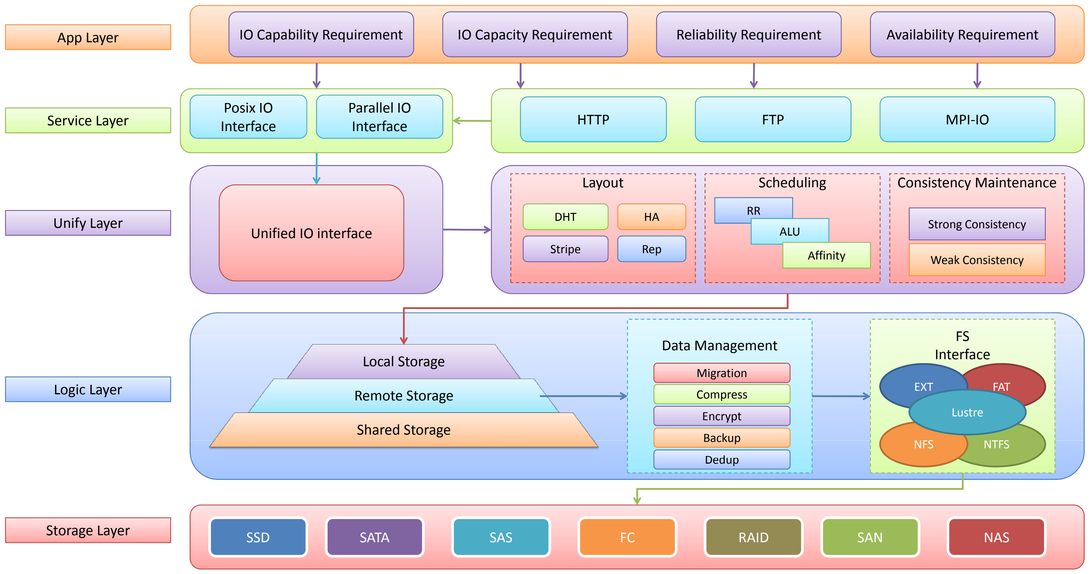
Large-scale Hybrid Tiered File System Architecture for the Post-petascale and Exascale computing

Prof. Yutong Lu (ytlu@nudt.edu.cn)
Lab of System Software
Department of Computer Science
National University of Defense Technology

Abstract

The scalability challenges faced by existing parallel file systems in HPC call for a new I/O architecture to meet the requirements of data-intensive scientific applications. With the increasing number of nodes in large-scale system, current shared I/O architecture suffer from serious problems such as performance variability, device contention, therefore impact on application performance. On the other hand local storage architecture is emerging as a remedy to current I/O architecture. We propose a new Large-scale Hybrid Tiered File System architecture (LHTFS), which can integrate shared storage, shared-nothing storage, Ram-disk, magnetic-disk and flash-disk into a multi-tiered storage architecture, and exploit benefits of hybrid to meet the performance and scalability gap between the compute and I/O in future large-scale system. We provide a unified and transparent view for users by utilizing syncretic management for different tiers of devices. The experimental results showed that LHTFS could effectively improve the overall performance of applications with intensive I/O.

File System Architecture



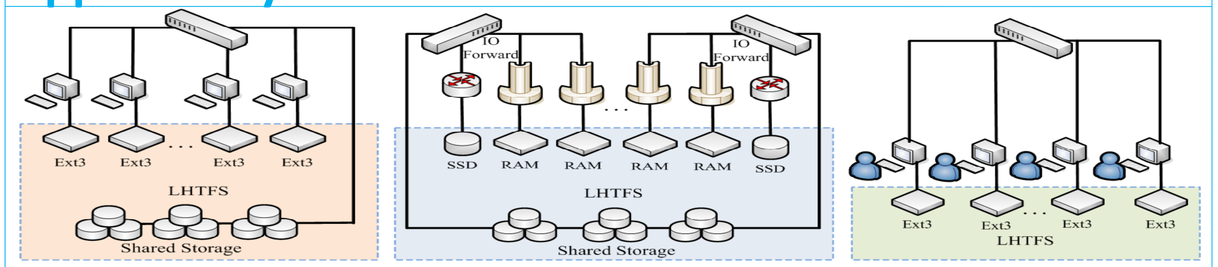
Objectives

- Capability to support post-petaflops and Exascale computing
- Scalability to achieve >1TB/s I/O bandwidth by leveraging spatial locality
- Applicability for supercomputers and clusters with hybrid infrastructure
- Usability by federating multi-level storage into unified name space
- Flexibility by key components re-configuration for application optimization

Research Background

- Development of parallel file systems for domestic YH/TH HPC systems
- THFS for TianHe-1A (Lustre-based)
  - Capacity: 2 PB, Scalability: clients>8192, oss>128
  - Performance improvement: Collective BW >100GB/s
    Customized protocol over proprietary interconnect, Optimized file cache policy
    Confliction release for concurrency accessing, Fine-grain distributed file lock
  - Reliability enhancement
    Fault tolerance of network protocol, Data objects placement, Soft-raid
  - Usability upgrade
    One-click installation, Maintenance, Zoning management

Applicability

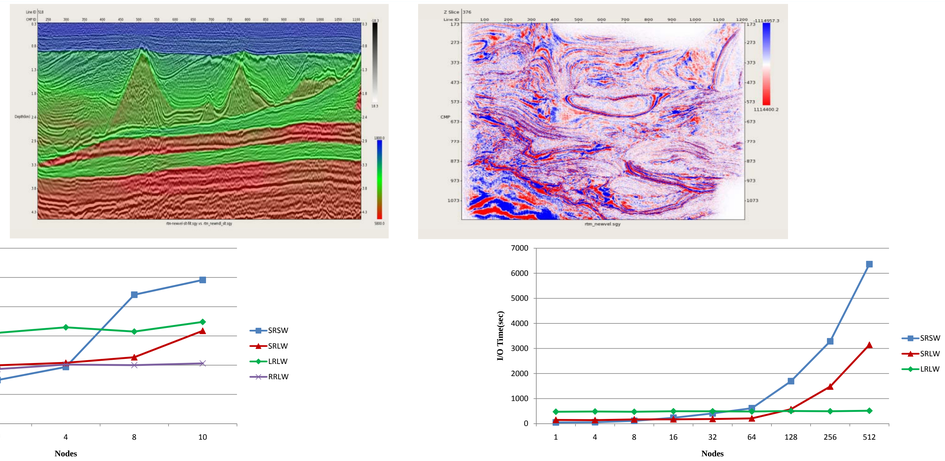


- Supports multiple tiers of storage devices and various configurations
- Exploits the features of local storage to improve the overall I/O performance in the case that both local storage and shared storage exist in HPC system
- Achieves IO forwarding by modifying node settings in the complex system configuration
- Acts as a normal distributed File System in the system without shared storage

Key Technologies & API

- Affinity scheduling to explore full potential of spatial locality
- Combination of centralized and decentralized metadata management
- Zoning and connecting on demand to reduce resource contention
- Relaxed data consistency control mechanism
- Vertical optimization through I/O path
- Redundant data management
- Smart data migration across storage levels
- POSIX-compliant UNIX file system interface for ease of use
- Custom API to boost app I/O access with weak consistency
- SDK for third-part interface support

Experimental Results



- Petroleum seismism data analysis: Typical DISC Application
- LHTFS used, Double tiers(Local disks + shared THFS)
- SRLW can reduce the overall IO time by around 50% than traditional SRSW
- Scalable I/O capability, LRLW I/O time is almost a constant

Open Issues

- Creative file system architecture for Exascale computing
- Effective & flexible meta-date management mechanism
- Large scale applications with complex intensive I/O