



An OSD-based Approach to Managing Directory Operations in Parallel File Systems

Nawab Ali^{#1}, Ananth Devulapalli^{*2}, Dennis Dalessandro^{*3},
Pete Wyckoff^{*4} and P. Sadayappan^{#5}

[#]The Ohio State University
^{*}Ohio Supercomputer Center

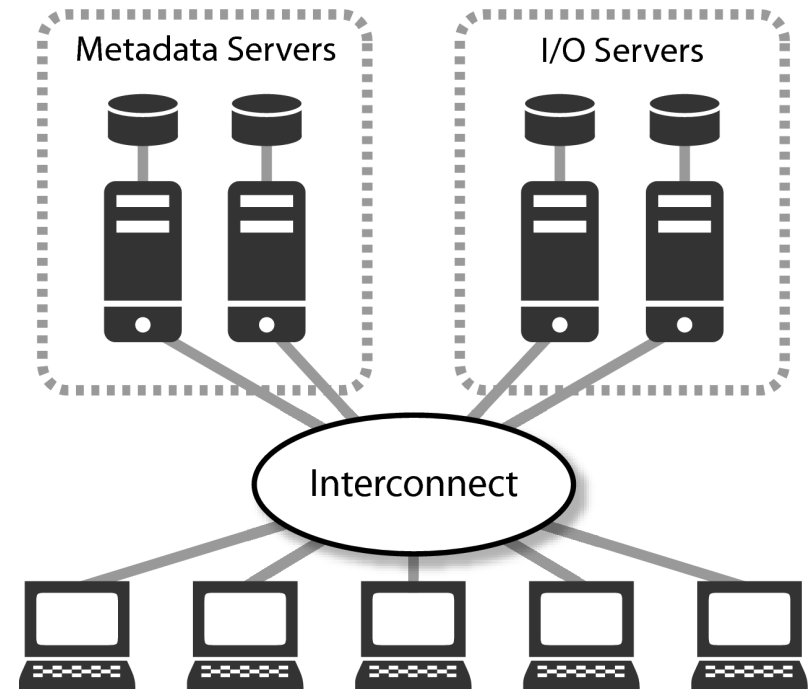
Presentation Outline

- Introduction
- Object-based Storage Devices
- System Design
- Representing Directories on OSDs
- Process Mutual Exclusion
- Experiments
- Conclusions & Future Work

- HPC applications increasingly generate or process large data sets
 - Sloan Digital Sky Survey
 - Large Hadron Collider
 - Climate Research
- Existing parallel file systems unable to cope with the I/O throughput requirements
 - Server-oriented design inhibits high-performance

Parallel File System Design Limitations

- I/O bandwidth and latency limit file system performance
- Store-and-forward latency
- Dedicated I/O and metadata servers limit
 - Scalability
 - Manageability
 - Performance

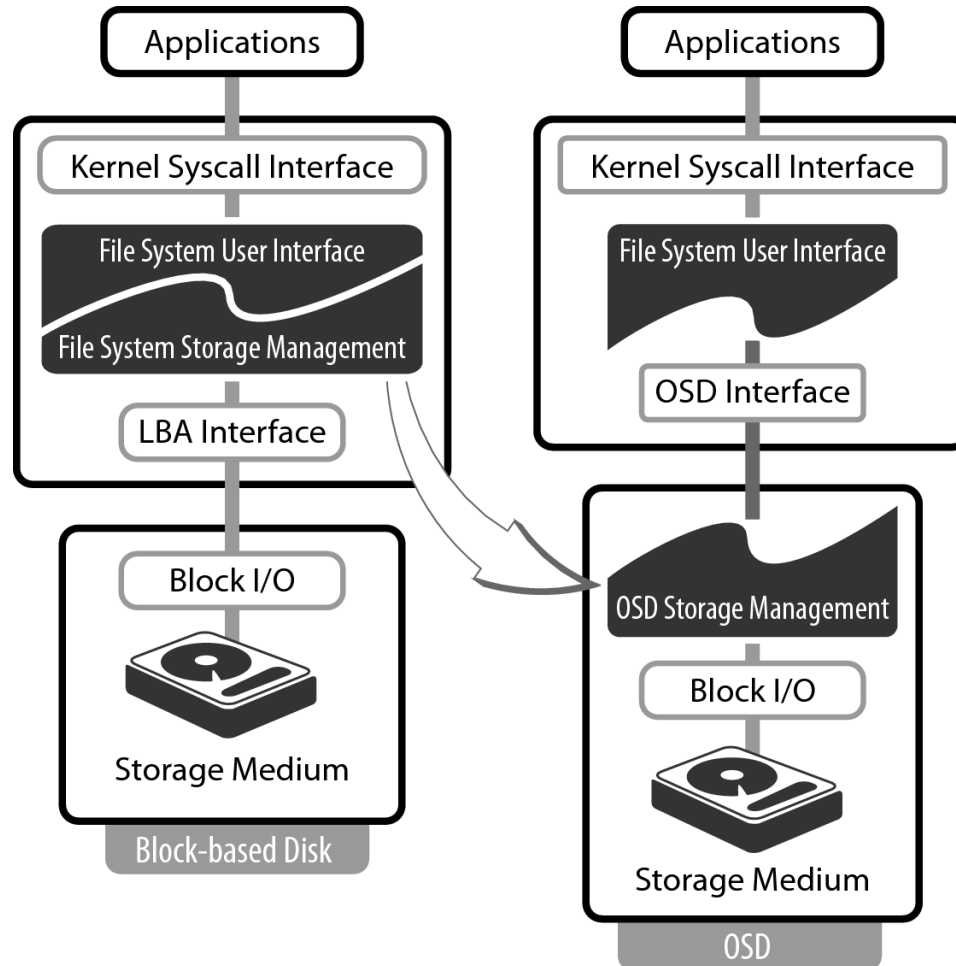


Typical parallel file system design

Object-based Storage Devices

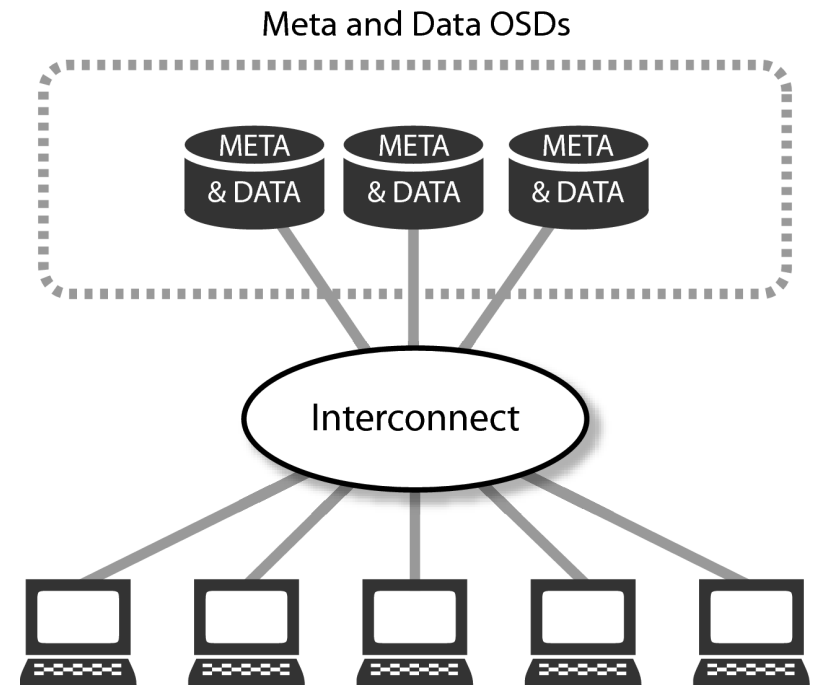
- New storage technology
 - SCSI extension
- Intelligent, higher-level object interface
 - Object encapsulation
 - Attributes
 - User assigned, but device managed
 - Large space for rich metadata
- Secure building block for direct-access file systems

OSD Architecture



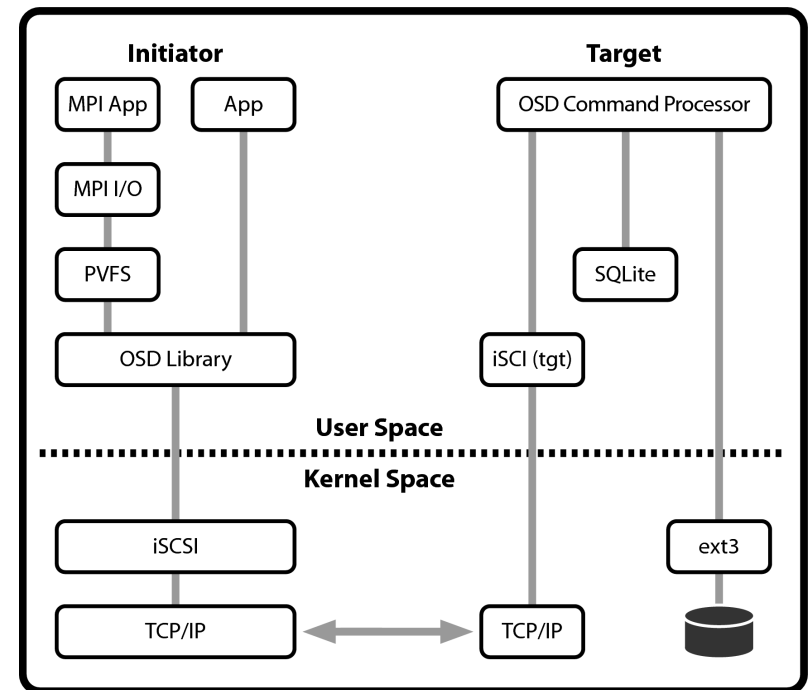
Parallel File System Design Goals

- Use intelligent peripherals (OSDs) to improve performance, scalability and manageability
- Serverless, direct-access storage model
- Recoupling data and metadata



OSD-based parallel file system design

- OSD Initiator
 - Exports OSD interface to client applications
 - Generates SCSI commands
- OSD Target
 - Software Implementation of OSD
 - OSD Command processor
 - Data Management
 - Attribute Management



OSD-based Directory Operations

- Parallel file systems provide concurrent access to multiple clients
- To ensure file system correctness, directory operations must be atomic
- Directory operations are challenging because of lack of atomicity in OSDs
- Propose extensions to the OSD T-10 standard
- Add support for compare-and-swap (CAS) operation

Representing Directories on OSDs

- Directory entries as object data
 - Directory entries are stored as file data
 - Directory operations are performed using file I/O
 - Slow and inefficient
- Directory entries as object attributes
 - Entries stored as user-defined attributes of the directory object
 - Directory operations are performed by manipulating object attributes

Process Mutual Exclusion

- Mutual exclusion among processes is required for manipulating directory entries in parallel file systems
 - Ensure uniqueness of dirents
 - File creation
 - Consistency of data structures
 - File removal

- SCSI Reservations
- Object Lock
- Atomic Operations
 - Device-based Compare-and-Swap (CAS)
 - Based on user-defined attributes

Locking Algorithms

- Lock-based Directory Access Protocol
- Atomic Directory Access Protocol

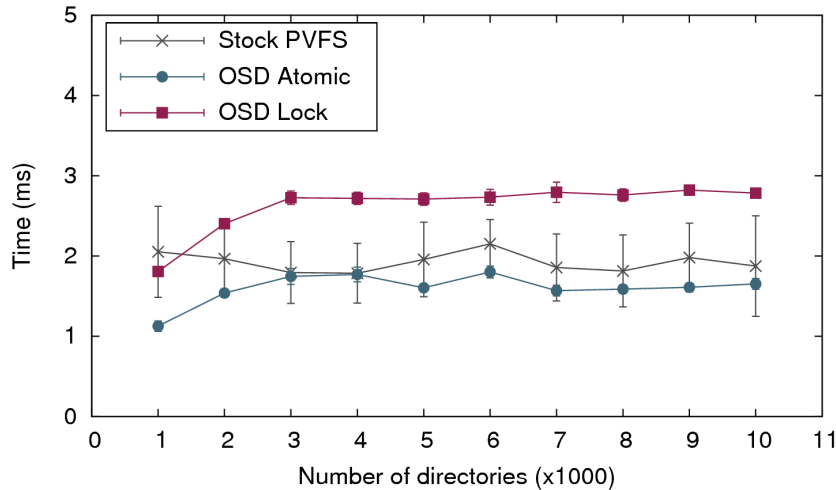
Lock-based Directory Access

- Based on conventional CAS
- Employs OSD LOCK operation to implement multi-client directory access
 - Lock directory
 - Perform lookup
 - Insert/remove
 - Unlock directory
- Simple and easy to implement
- Significant network overhead (4 RTTs)

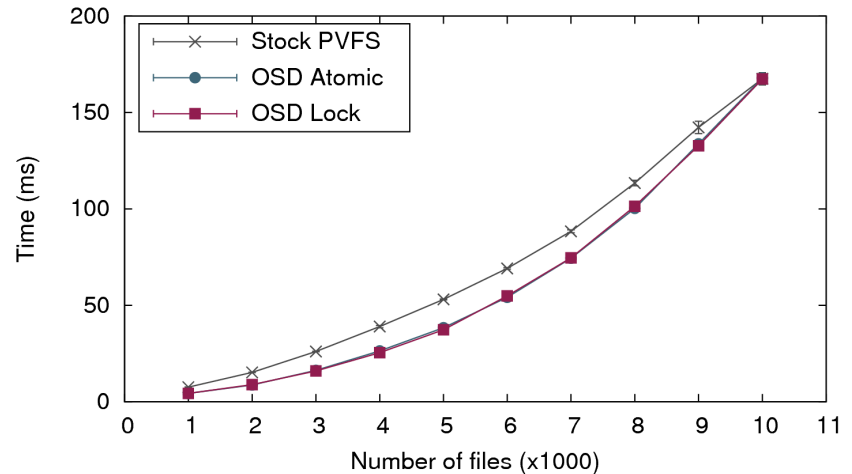
Atomic Directory Access

- Based on generic CAS operation
 - Compares an array of bytes as opposed to integer
- Reduces insert/remove to a single message exchange
 - Insert: CAS(NULL, dirent)
 - Remove: CAS(dirent, NULL)
- Mitigates the high network overhead of lock-based protocol

Latency Microbenchmarks



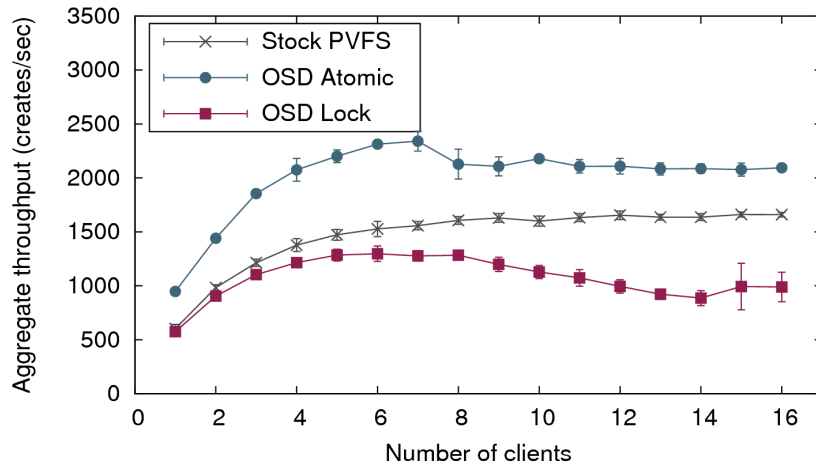
Mkdir



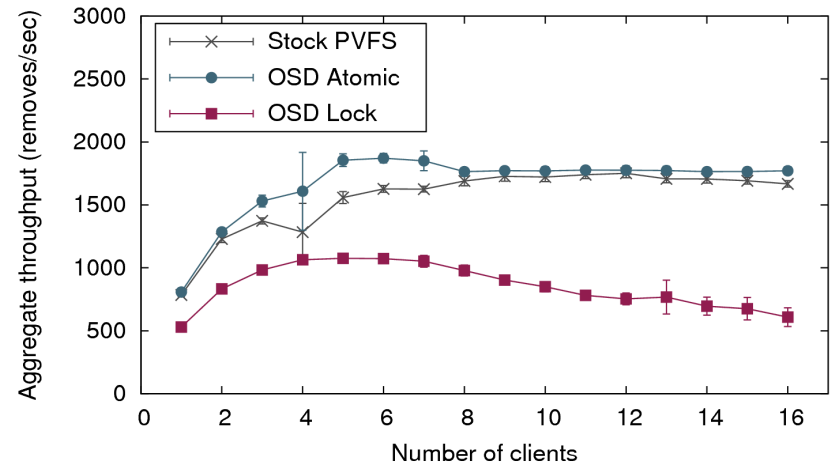
Readdir

- Latency of dirops as a function of number of objects
- 1 client. 1 PVFS server or 1 OSD respectively
- Mkdir: OSD Atomic has lower latency than PVFS and OSD Lock
- Readdir: Latency dominated by network transmission time

Throughput Microbenchmarks

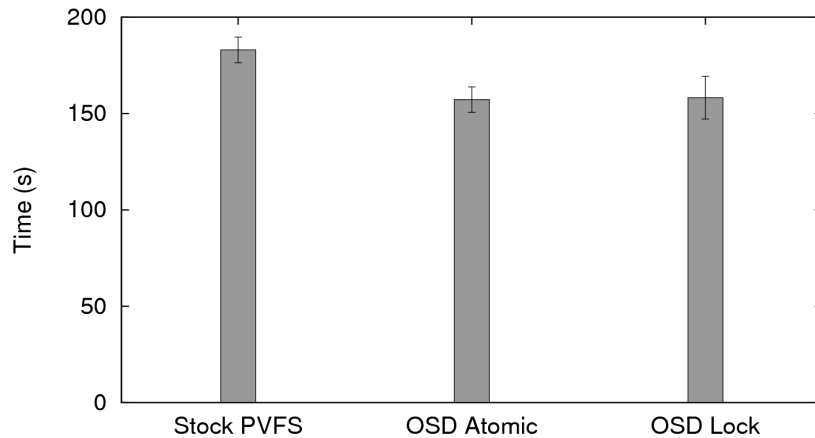


Create

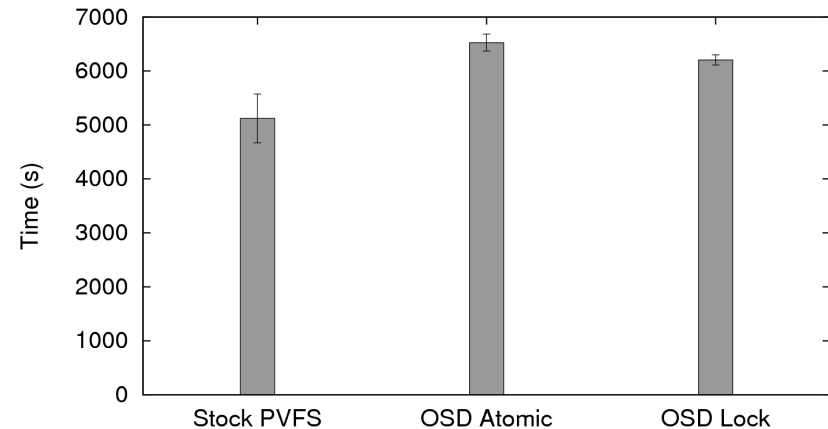


Remove

- Throughput of dirops as a function of number of clients
- 4 PVFS servers or 4 OSDs respectively
- Create: OSD algorithms do not create metafile object
- Remove: User-defined attrs need to be removed before object deletion



SSCA-3 test1



SSCA-3 test7GB

- 1 client. 4 PVFS servers or 4 OSDs respectively
- test1: 5000 files. 1 GB data. Small metadata footprint
 - Execution time dominated by small I/O
- test7GB: 95000 files. 7 GB data. Metadata intensive

- OSD-based parallel file system obviates the need for dedicated servers
- Implemented directory operations on OSDs
- Proposed extensions to the OSD T-10 standard
 - Compare-and-Swap
- Presented 2 OSD-based directory access protocols
- Performance of OSD-based file system is comparable and in some cases better than stock PVFS

- Metadata query based on user-defined attributes
- File system scalability analysis
 - OSC Glenn cluster
 - iSER
- Replicated Metadata

Acknowledgment

- This material is based upon work supported by the National Science Foundation under Grant No. 0621484

Thank You