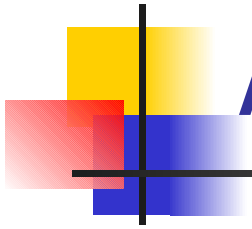
# Coordinating Parallel HSM in Object-based Cluster Filesystems

Dingshan He, Xianbo Zhang, David Du
University of Minnesota

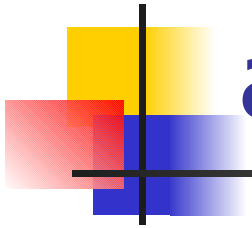
Gary Grider
Los Alamos National Lab



# Agenda

- Motivations
- Parallel archiving/retrieving architecture
- Coordinating parallel hierarchical storage management
- Experiment result
- Future works

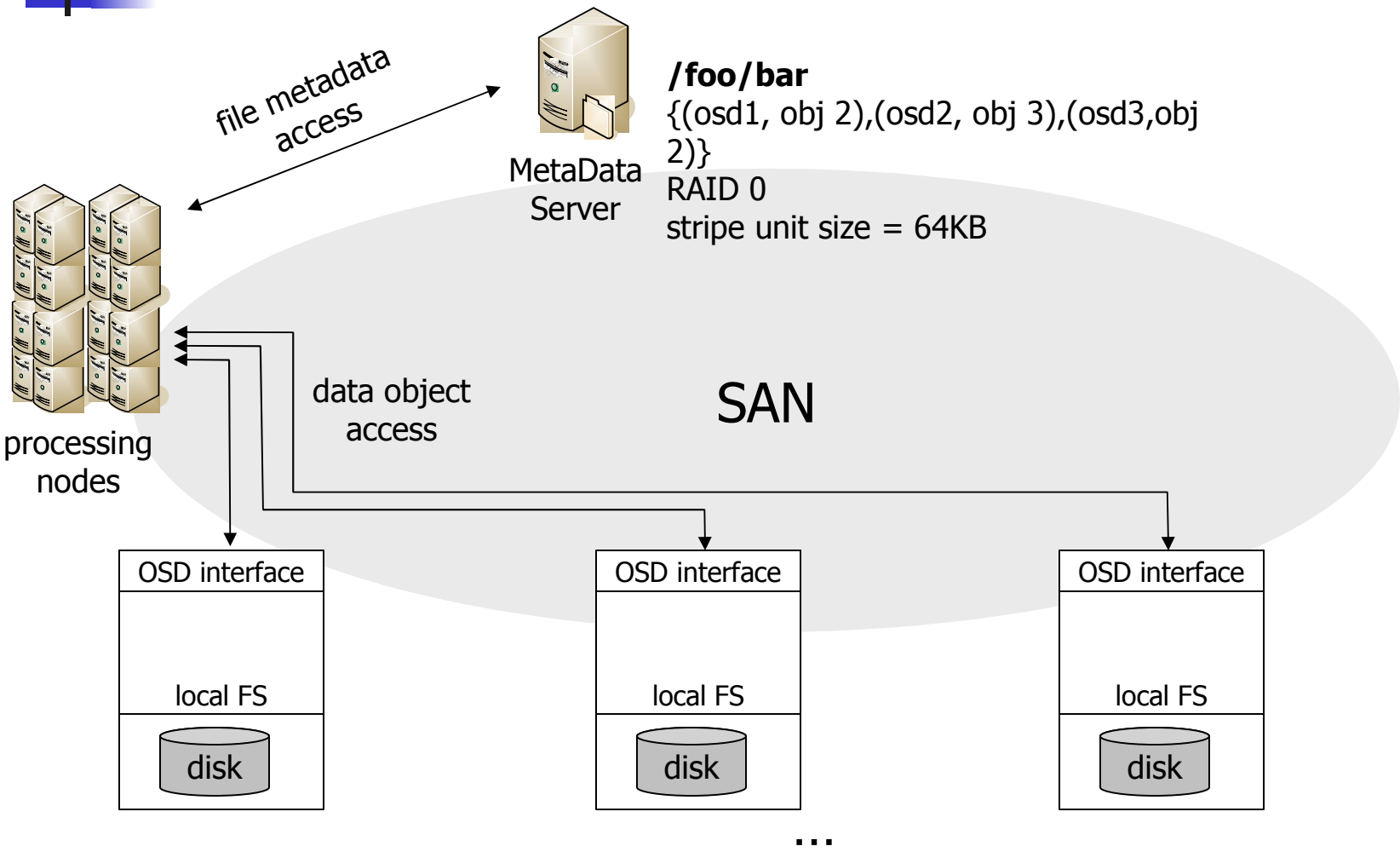
# Demanding for Scalable, Global and Secure (SGS) file system

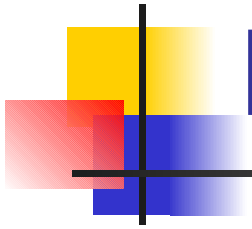


- High Performance Computing (HPC) applications
- Tri-Lab File System Path Forward RFQ
  - Global name space
  - Security
  - Scalable infrastructure for clusters and enterprise
  - No single point of failure
  - POSIX-like Interface
  - Work well with MPI-IO

  …

# Object-based Cluster File System (OCFS)

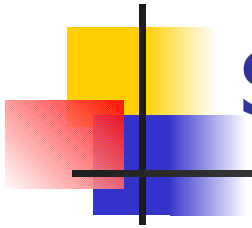




# Recent OCFS Solutions

- Lustre File System of CFS, Inc.
  - LLNL runs Lustre on Multiprogrammatic Capability Cluster (MCR)
  - 20 million files and 115.2TB
  - Aggregate I/O 22GBps
- ActiveScale File System of Panasas
  - LANL deploys three systems
  - The largest one has 200TB capacity and ~20GBps aggregate I/O

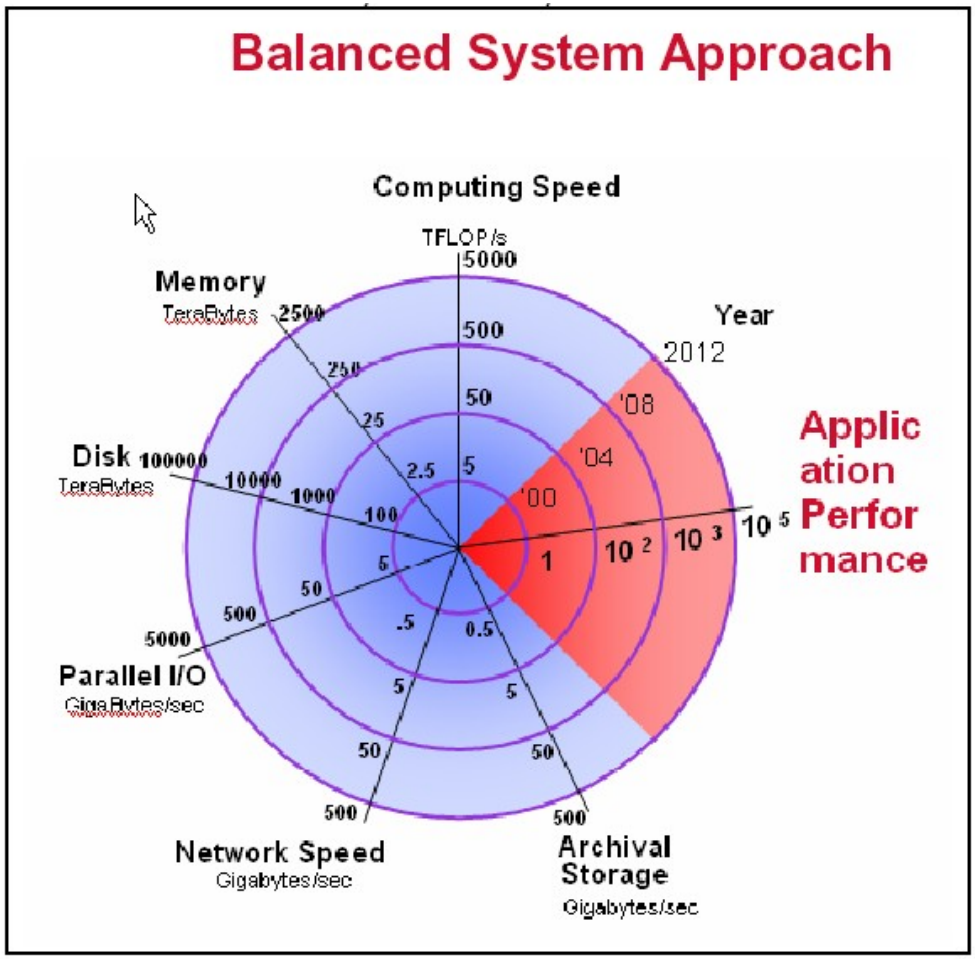
# Why storage hierarchy in SGS systems?

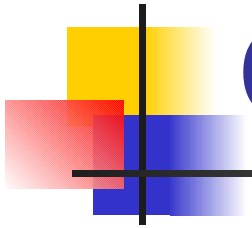


- Fast data generating rate in HPC environment
  - simulations generate one new file of multiple Gigabytes every 30 minutes
  - Checkpoints for *error recovery* and *experiment steering*
- Cost effectiveness
  - Combination of expensive high-performance storage and more affordable low-performance storage
  - Data lifecycle
  - Shared computing facilities used by many projects in turns

# Bottleneck in archive and retrieve bandwidth

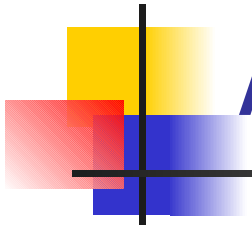
- Enlarging gap between application parallel I/O and archival storage I/O
  - Every TFLOP needs 1GBps parallel I/O bandwidth
  - Every TFLOP needs 100MBps archival I/O bandwidth
  - 2005 BlueGene/L 280TFLOP
- Backup and Restore record in 2003
  - Achieved by SGI
  - 2.8GBps file-level backup
  - 1.25GBps file-level restore





# Objectives

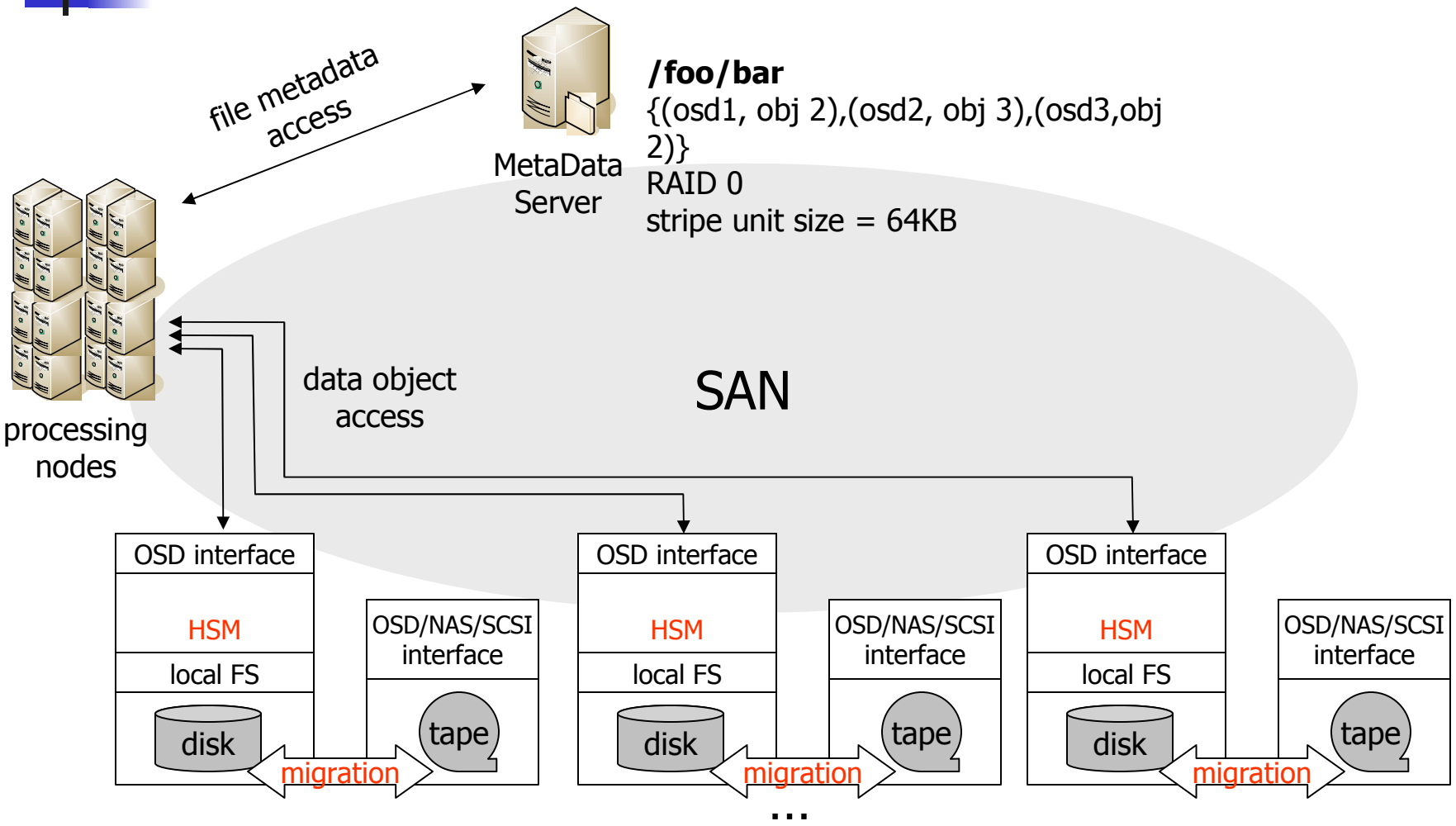
- High aggregated data archive and retrieve bandwidth
- Scalable in archival bandwidth in addition to capacity
- Automated and transparent management of data migration in storage hierarchy

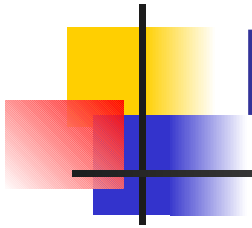


# Agenda

- Motivations
- Parallel archiving/retrieving architecture
- Coordinating parallel hierarchical storage management
- Experiment result
- Future works

# Parallel Archive Architecture



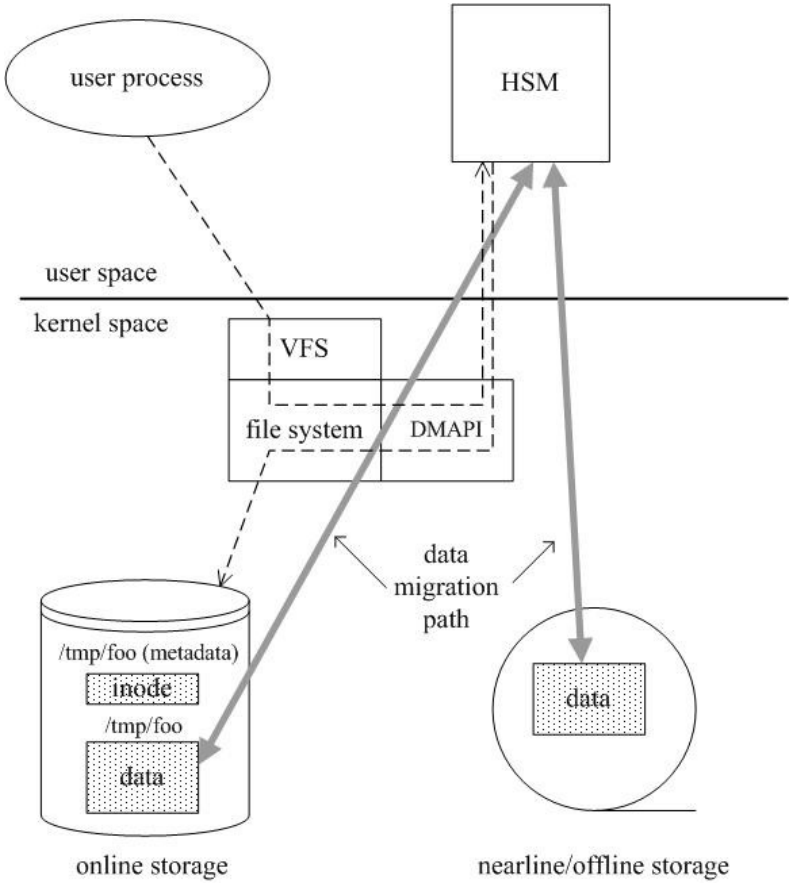


# Design Rationales

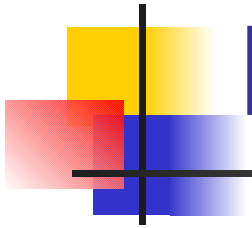
- Parallel archival storages
  - Explore aggregated parallel archival bandwidth
- OSD embeds automated management of migrations
  - Close to storage device
  - Better understanding of object access pattern
- Direct data migration between OSDs and their associated archival storages
  - OSDs are smart and powerful enough

# Eliminating dependency on Data Management API (DMAPI)

- Not widely supported by popular file systems
- Not scalable from past experiences
  - Requiring single event listener
- Most functions are unused by HSM

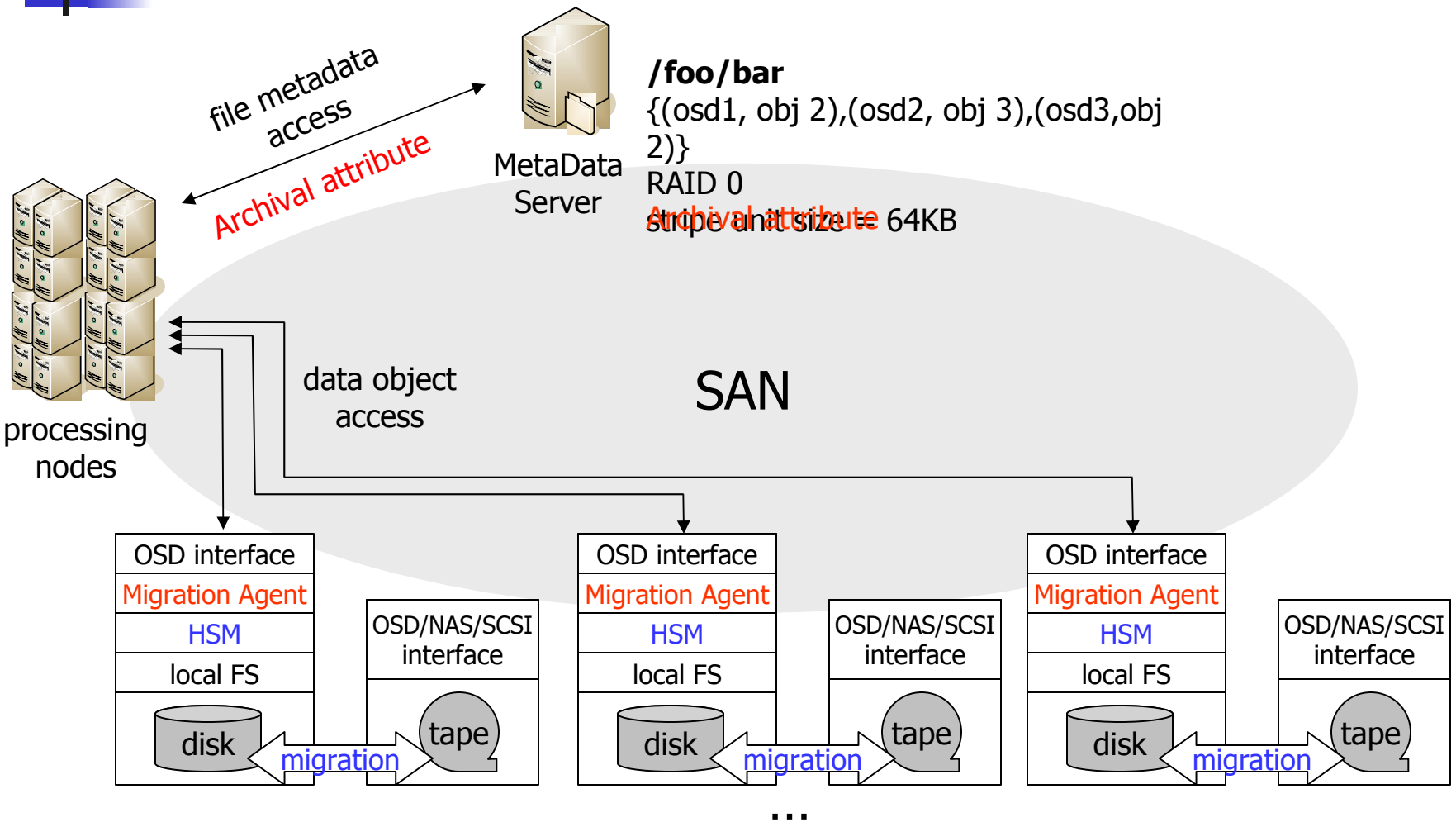


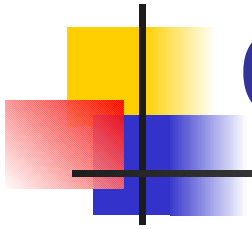
# Functions of DMAPI Need to be Replaced



- Catching access events
  - Accessing objects not in online storage
  - Triggering HSM to recall data
- Transparent namespace
  - As if files are always there
  - File stub is always kept in the FS managed by HSM

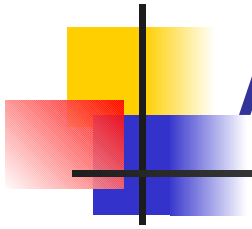
# Replacing DMAPI





# Component Functions

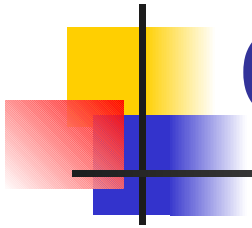
- Archival attribute
  - File object storage level
  - File object location
- Migration Agent
  - Intercepting file system requests from object interface
  - Checking archival attribute to pass the request to local file system or recall object first



# Agenda

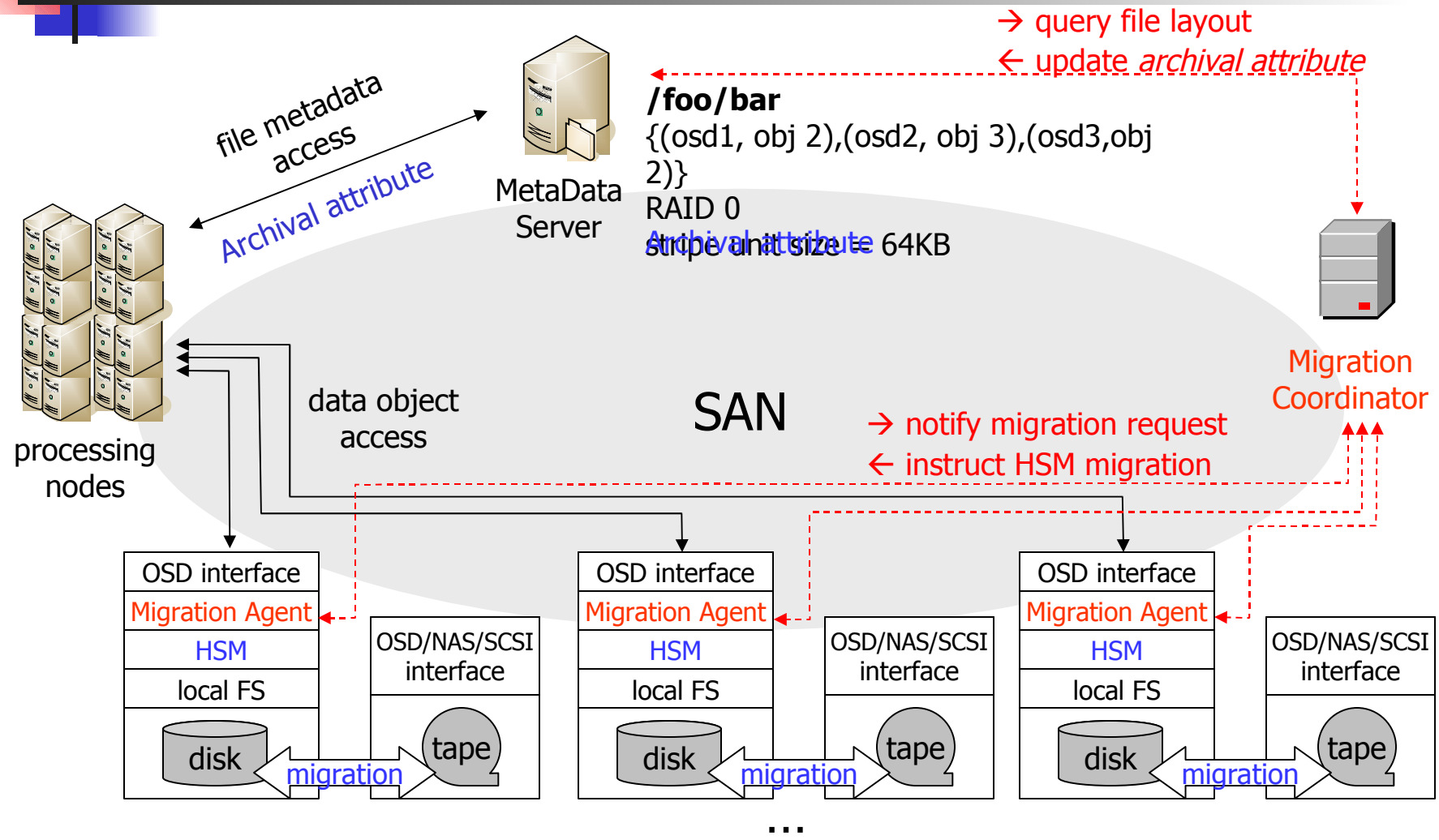
- Motivations
- Parallel archiving/retrieving architecture
- Coordinating parallel hierarchical storage management
- Experiment result
- Future works

# Distributed HSMs Need to be Coordinated

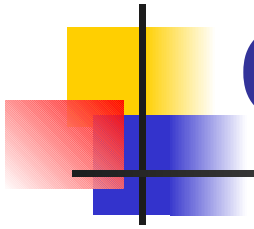


- Striping of single large files on multiple OSDs
  - Multiple gigabyte files are common in HPC
- File sets of many related files on multiple OSDs
  - Used by the same application
- "Synchronous" migrations between multiple pairs of online OSD and archival storage
  - True high aggregated migration bandwidth for single file or file set
- Sequential access pattern can not explore parallel migration data paths
  - Striped file will be retrieved object by object in sequence

# Coordinating Parallel HSM

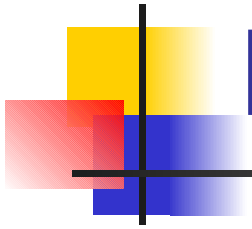


# Reasons for Centralized Coordinating

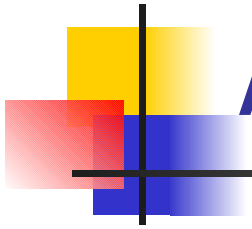


- Separated migration control path and migration data path
  - OSDs do not involve in distributed migration coordinating task
- Centralized coordinating authority
  - Possible for intelligent decisions across requests

# Concurrency Control and Error Recovery



- Migration Locking Mechanism
  - Concurrent client access, archive migration and retrieve migration
  - Access Lock (ALock); Backup Lock (Block); Restore Lock (RLock)
- Error Recovery
  - Failure on components in the middle of migration
  - Logging migration operation before starting
  - Checkpoint for large object migration
  - Restart unfinished migration operation when doing error recovery

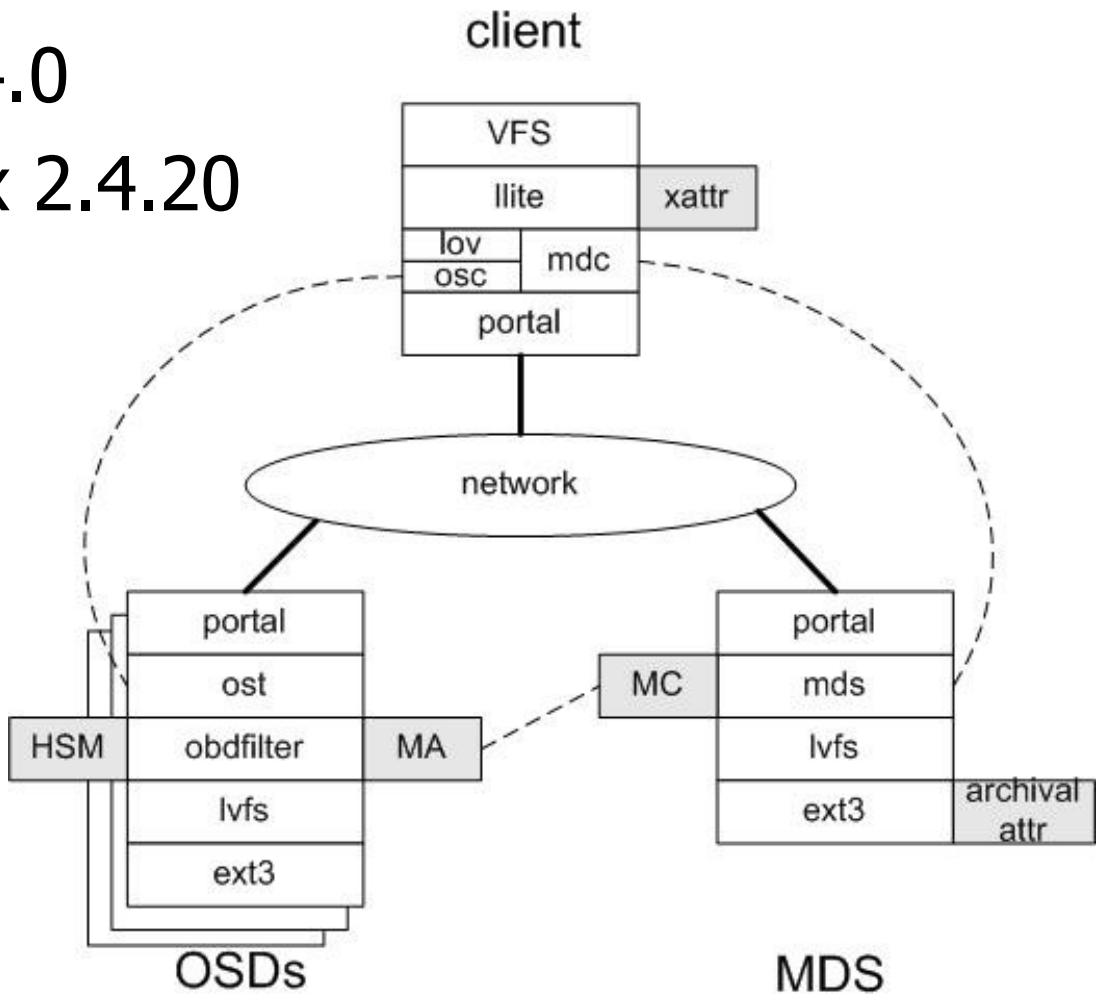


# Agenda

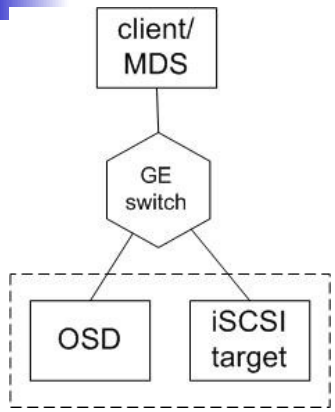
- Motivations
- Parallel archiving/retrieving architecture
- Coordinating parallel hierarchical storage management
- Experiment result
- Future works

# Prototyping

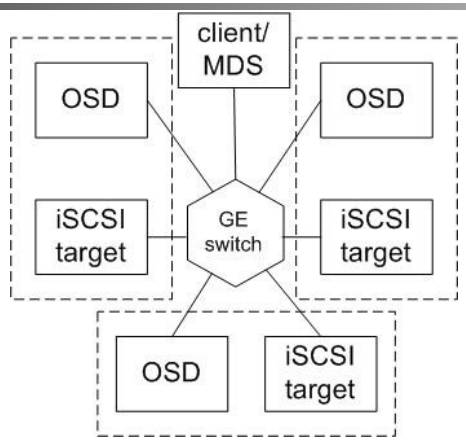
- Lustre 1.4.0
- RH9 Linux 2.4.20



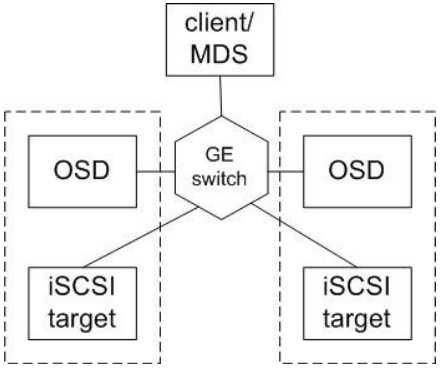
# Experiment setup



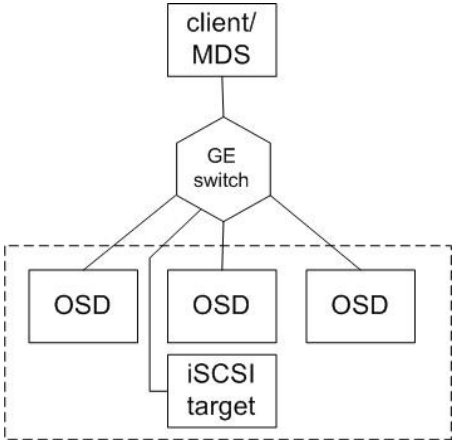

single-pair




triple-pair




dual-pair




single-backup

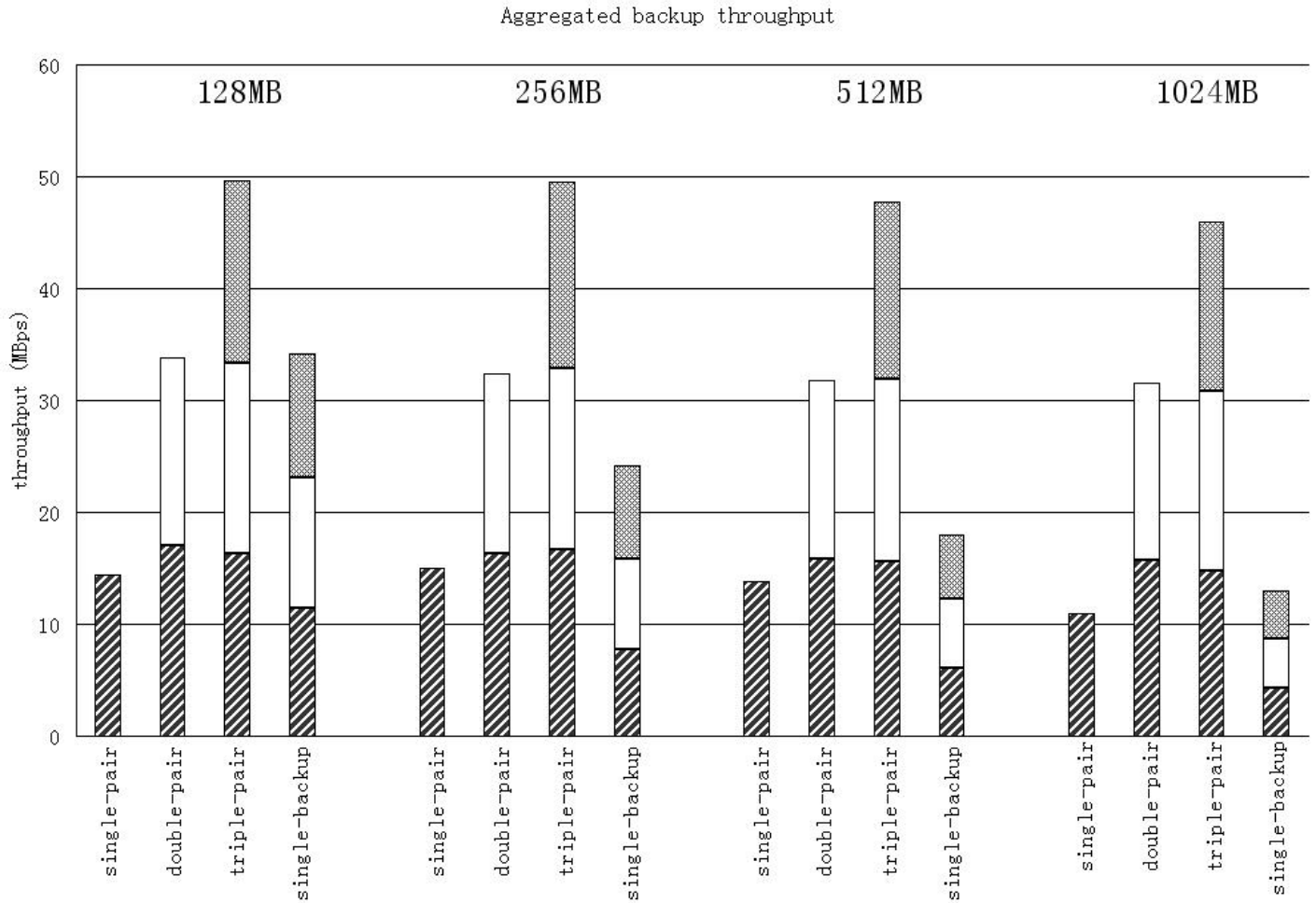
## OSD host configurations

| | |
|---|---|
| CPU | Two Intel XEON 2.0G |
| Memory | 256MB DDR DIMM |
| SCSI interface | Ultra 160 SCSI (160MBps) |
| HDD speed | 10,000RPM |
| Avg. seek time | 4.7ms |
| NIC | Intel Pro/1000MF |

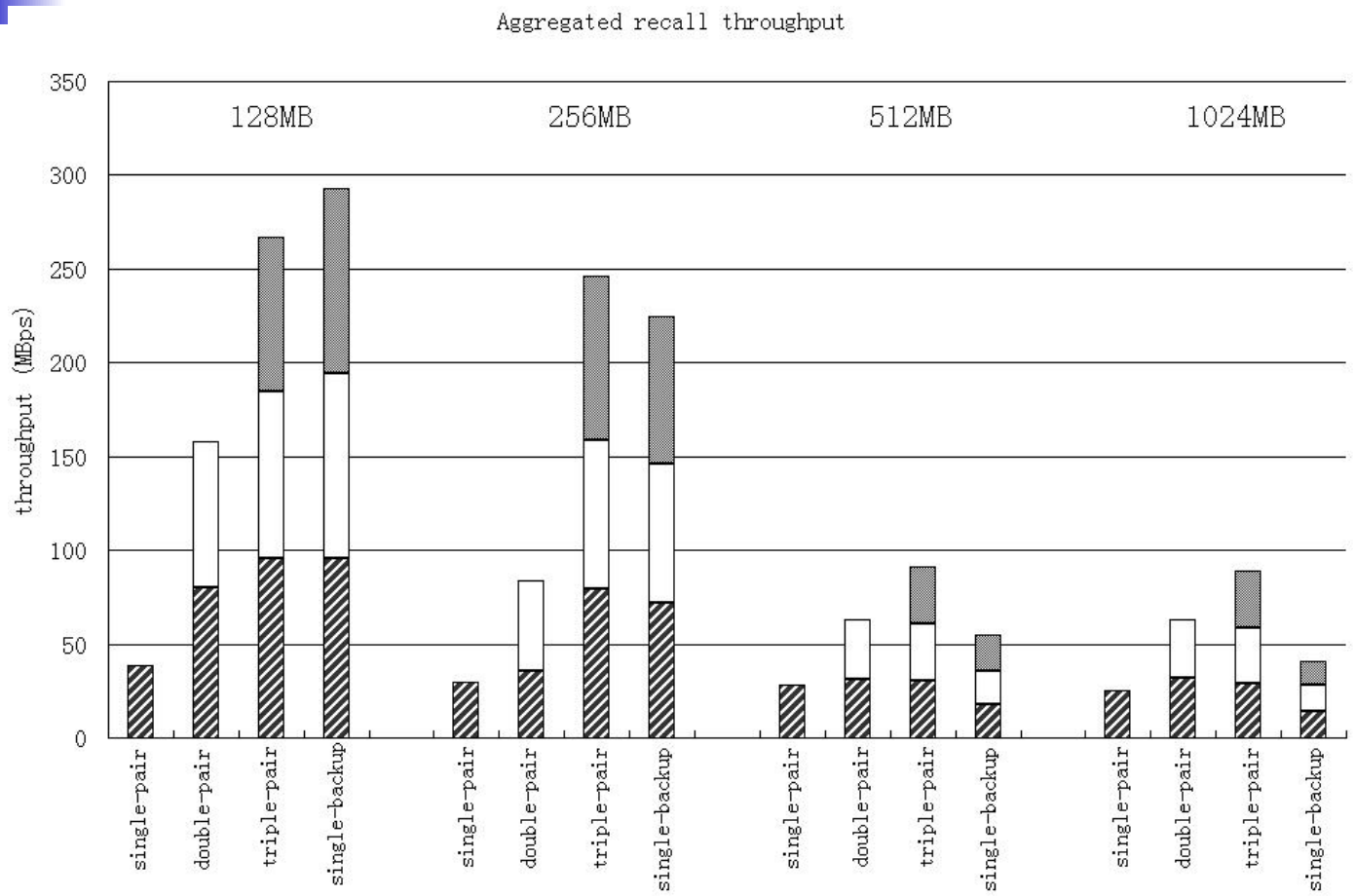
## Target/MDS host configurations

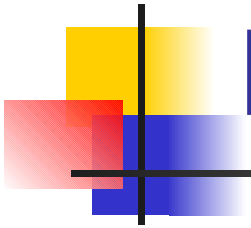
| | |
|---|---|
| CPU | 4 Pentium III 500MHz |
| Memory | 1GB EDO DIMM |
| SCSI interface | Ultra2/LVD SCSI (80MBps) |
| HDD speed | 10,000RPM |
| Avg. seek time | 5.2ms |
| NIC | Intel Pro/1000MF |

# Aggregate Archive Bandwidth



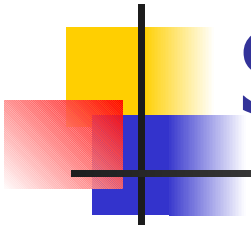
# Aggregate Retrieve Bandwidth





# Future Works

- Intelligent migration decision
  - How to use both file metadata access pattern from MDS and file object access pattern from OSDs to make better migration decisions
- Object archive tertiary storage devices
  - Device interface for archive and retrieve operations
  - Format of portable storage media
  - Sequential-access features of tapes
- Extension to pNFS
  - Handling heterogeneous storage device types



# Summary

- A hierarchical storage solution for Object-based Cluster File Systems
- High aggregate archiving/retrieving bandwidth
- Scalable architecture for both archival capacity and bandwidth
- Transparent and automatic object migrations

# Acknowledgements


- Los Alamos national lab


- DTC Intelligent Storage Consortium (DISC) members


  - ITRI
  - LSI Logic
  - Sun Microsystems
  - Symantec




# Sequence of Accessing Object not in online storage