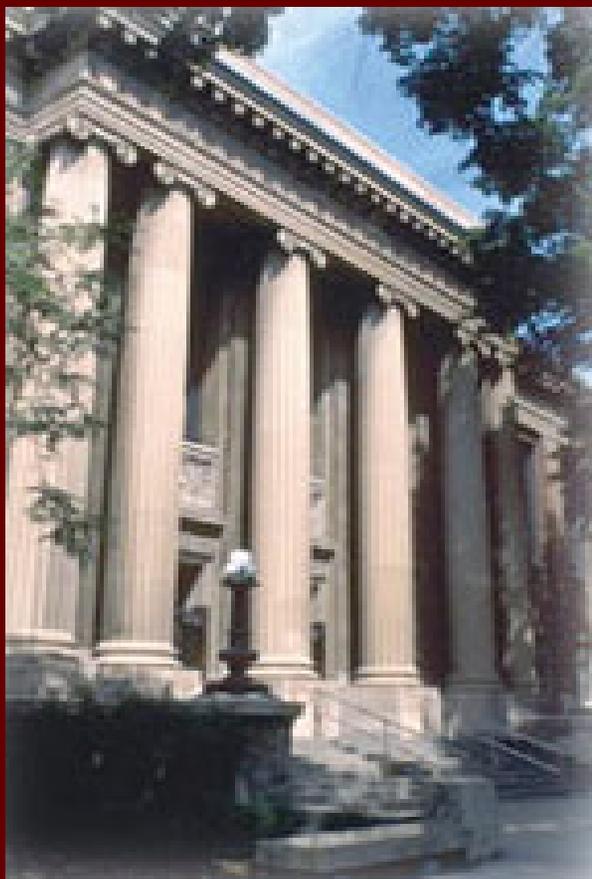


# Integrated Infrastructure for Secure and Efficient Long-Term Data Management



**PI: Andrew Odlyzko**  
**co-PI: David Lilja, Yongdae Kim**

## DISC

**University of Minnesota**  
*Digital Technology Center*  
*Intelligent Storage Consortium*

**David DU**

# Outline



- Overview of Proposed Research
- Progress to date
- Wish-list
- Publications
- Involved Students

# Motivations



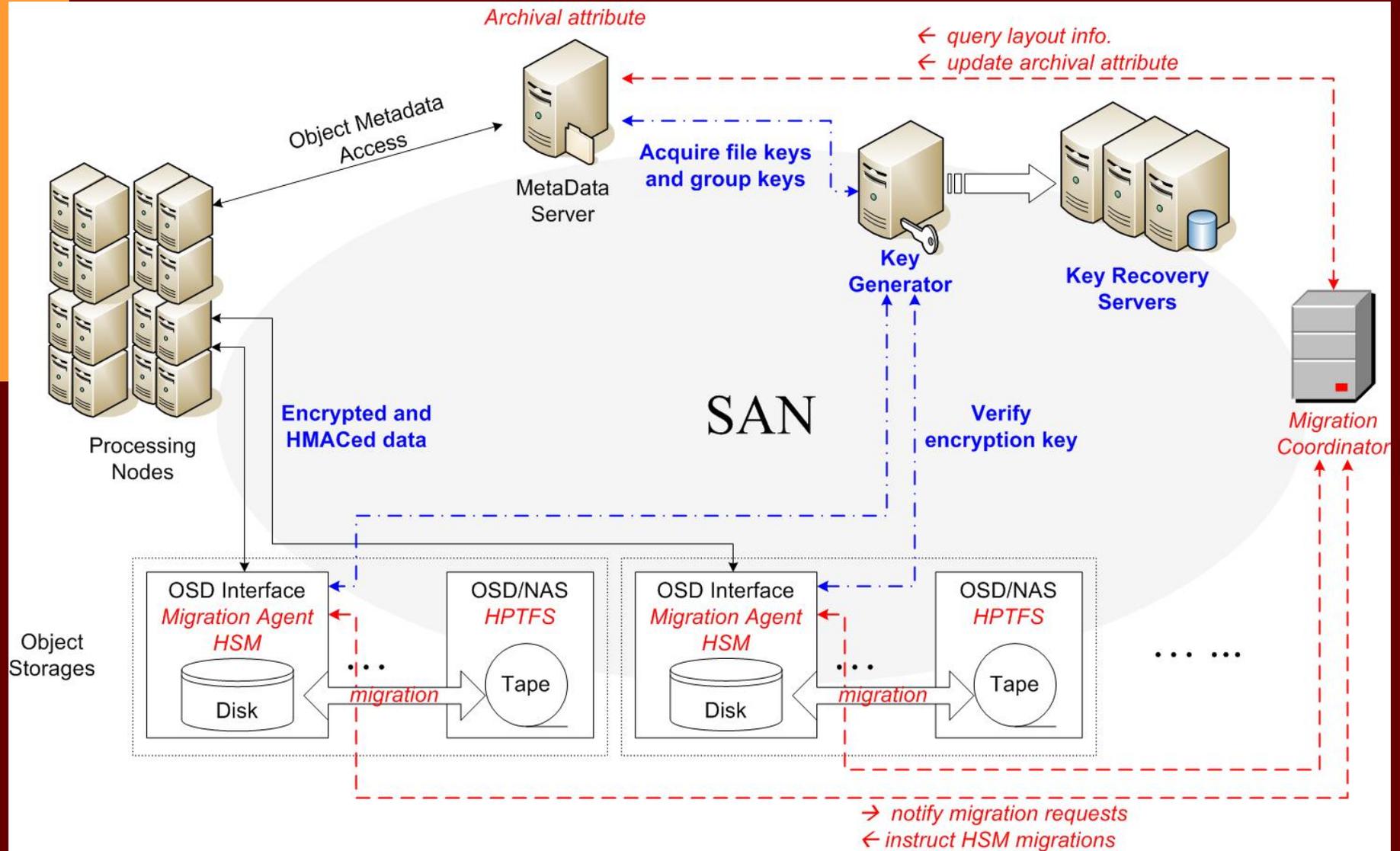
- Long-Term Data Preservation Becomes Extremely Important for Digital Data
- Data Backup and Data Security Are Two Critical Issues for Long-Term Data Preservation
- Data Encryption Is A Must for Data Protection
- However, **Long-Term Key Management** Is A Challenge
- **Efficient Data Backup and Archive** Is An Essential Part of Long-Term Data Preservation

# Requirements and Focus



- Requirements for Long-term Data Archiving and Protection
  - High data rate archive and restore throughput
  - Automated and transparent management of data migrations in storage hierarchy
  - Efficient backup and retrieval of keys
  - Key recovery
  - Long-term management
    - ▣ group reorganization such as creation/deletion/split/merge
  - Usability
  - Scalability
- Current Focus of Project:
  - Transparent backup and archive functions
  - High-performance backup, restore, and data access operations
  - Efficient techniques for ensuring long-term data security and accessibility

# System Architecture



# Data Archiving



- Local HSM agent on OSD
  - automates the data migration between the OSD's internal storage and a designated archival storage on the SAN
  - allows parallel data migration paths to achieve high aggregated migration throughput
- Migration Coordinator
  - initiates parallel data migrations to take advantage of the parallel data paths provided by the physical topology
  - guarantees the consistent archiving state of a set of related data objects
  - helps to eliminate heavy loaded DMAPI
- High Performance Tape File System (HPTFS)
  - eases the sharing and usages of tape libraries as archival storages
  - enables accessing tape-based archival storage using either OSD interface or NAS interface.

# Long-Term Key Management



- Securing data at rest
  - End-to-end encryption = Writer encrypts, reader decrypts
  - Previous key management work focused on providing solutions satisfying a single requirement
    - e.g. Hierarchical key management for improving scalability, Key rolling for efficient recovery of past keys, Broadcast encryption and group key distribution for efficient revocation
  - This projects investigate key management solutions that satisfy multiple requirements for long duration.
- Transparent encryption and long-term key management
  - to improve usability and manageability
- Key recovery and backup
  - Adopting and improving cryptographic key recovery mechanism for storage

# Blending Multiple Requirements



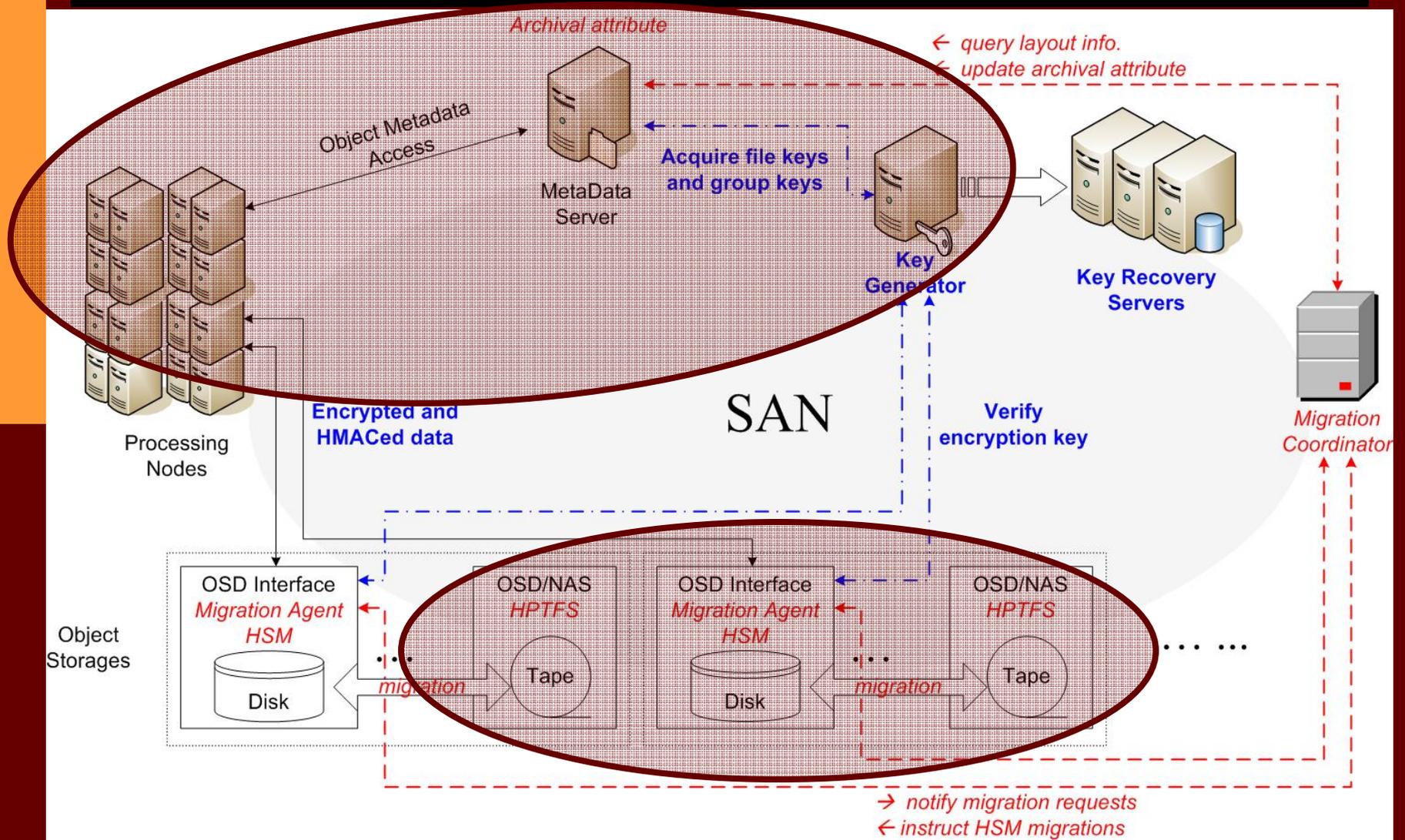
- Limited Roll-back
  - Previous solutions allow to roll-back indefinitely
    - ☐ Not necessarily secure for all environments
  - Can we limit the number of roll-backs so that new users might have access to data created in a specified period (without sacrificing significant performance)?
- Efficient Hierarchical Access Control
  - RBAC (Role-based Access Control) provides efficient grouping based on roles
  - Hierarchical key management may reduce number of keys managed by individual nodes
  - But, it fails to achieve similar efficiency as RBAC
    - ☐ i.e. revocation of higher-level node = revocation of all nodes under the high-level node
  - No effort in response to merge/split of groups in hierarchical key management

# Our Design Goal



- Hierarchical key management with limited key roll-back
  - Efficient key revocation
  - Response well with organization and individual changes
  - Good security and manageability balance
  - Guaranteed key recovery

# Progress-to-date



# Wish-list



- Information on Organization Changes and Data Creations/Deletions for Longer Term (50 to 100 years)
- Usage Patterns for Enterprise Customers

## Publications (OSD & Backup)



- X. Zhang, D. Du, J. Hughes and R. Kavuri, “HPTFS: High Performance Tape File System”, MSST 2006
- D. Du, et al, “Experiences on Building An Object-Based Storage System Based on T10 Standard,” MSST2006; **Source code is available for download on <http://sourceforge.net/projects/disc-osd/>**
- D. He, X. Zhang, D. Du and G. Grider, “Coordinating Parallel Hierarchical Storage Management in Object-Based Cluster File Systems,” MSST 2006

## Publications (Cont.)



- D. He, N. Mandagere and D. Du, “Design and Implementation of A Network-Aware Object-Based Tape Device”, MSST 2007 (accepted)
- N. Mandaere, J. Diehl and D. Du, “GreenStor: Application-Aided Energy-Efficient Storage” MSST 2007 (accepted)
- X. Zhang, D. Du, J. Hughes and R. Kavuri, “ITStorage: A High Performance Internet Tape Storage” submitted
- Y. Lu, D. Du and X. Zhang, “QoS Scheduling for Object Storage Systems” submitted

## Publications (Security)



- V. Kher, Y. Kim, “Building Trust in Storage Outsourcing: Secure Accounting of Utility Storage”, SRDS’07
- CoreFS: source code is available for download  
<http://www.cs.umn.edu/research/sclab/coreFS.html>
- EncFS in Preparation
- Pipelined Encryption in Preparation

# Publications (Storage & Database)



- A. Raghuvver, D. Du, M. Mokebel, B. Debnath, and M. Jindal, “SQUAD: An Intelligent-Storage Approach to Integrating Structured and Unstructured Data”, accepted by ACM Conference on Information and Knowledge Management 2007
- B. Debnath, J. Skarie, D. Lilja, and M. Mokbel, “SARD: A Statistical Approach for Ranking Database Tuning Parameters,” submitted to ICDE.
- J. Skarie, B. Debnath, D. Lilja, and M. Mokbel, “SCRAP: A Statistical Approach for Creating a Database Query Workload Based on Performance Bottlenecks,” IEEE International Symposium on Workload Characterization (IISWC), 2007.

# Involved Students



- PhD students
  - N. Mandagere, 3rd year
  - V. Kher, 5th year
  - B. Debnath, 2nd year
  - N. Park, 2nd year
  
- MS students
  - J. Skarie
  - S. Hong (graduated)



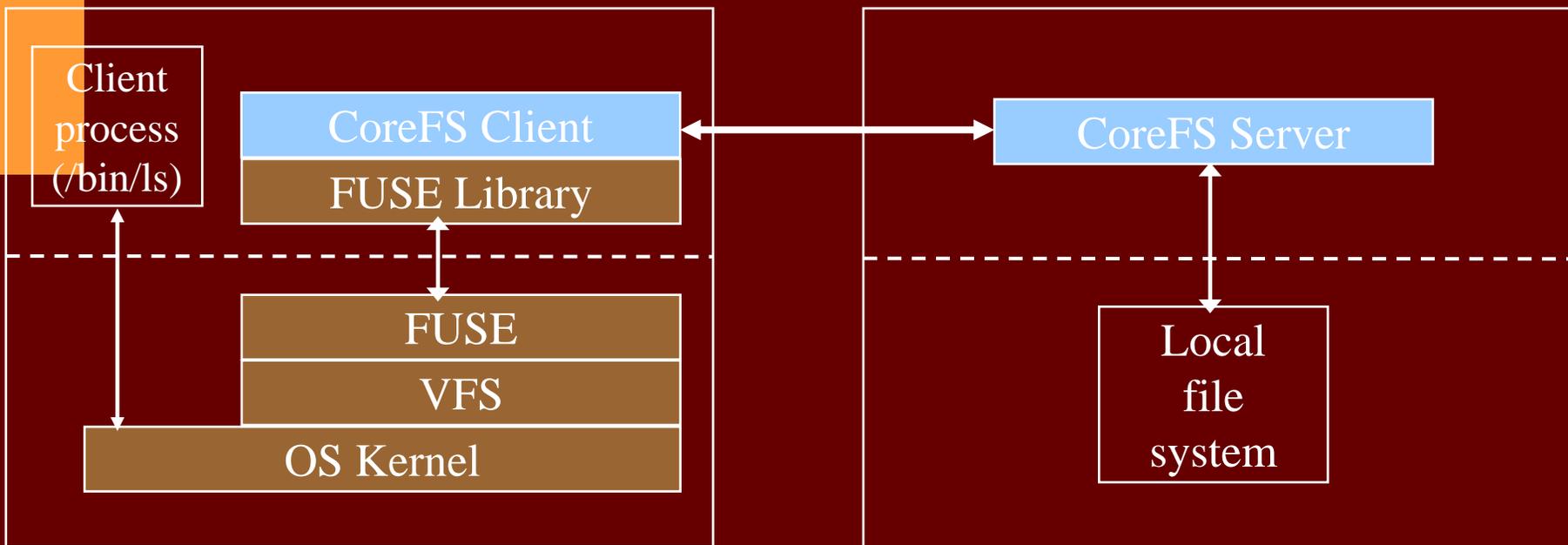
# Questions?



# CoreFS



- A Simple Network File System
- Implement only the “core” functionalities
  - Give developers a simple network file system
- Extend this file system with security modules



Download: <http://www.cs.umn.edu/research/sclab/coreFS.html>

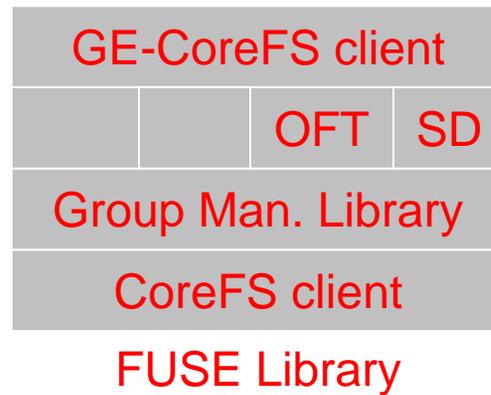
# EncFS



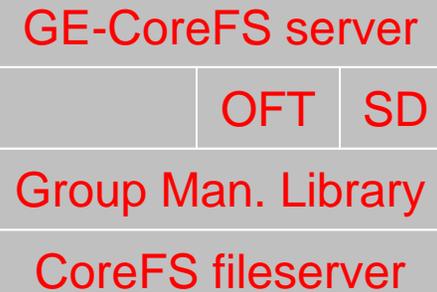
Key Server  
OFT SD  
Group Man. Library

Key Server  
Machine

suseradd id..  
suserdel id..  
  
vi topsecret.txt



FUSE kernel  
VFS



VFS

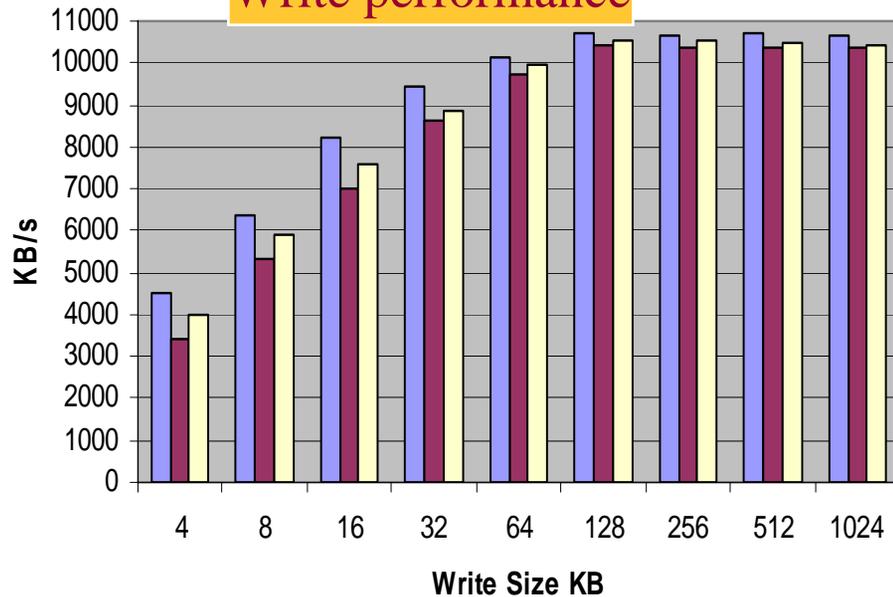
Client Machine

Storage Server

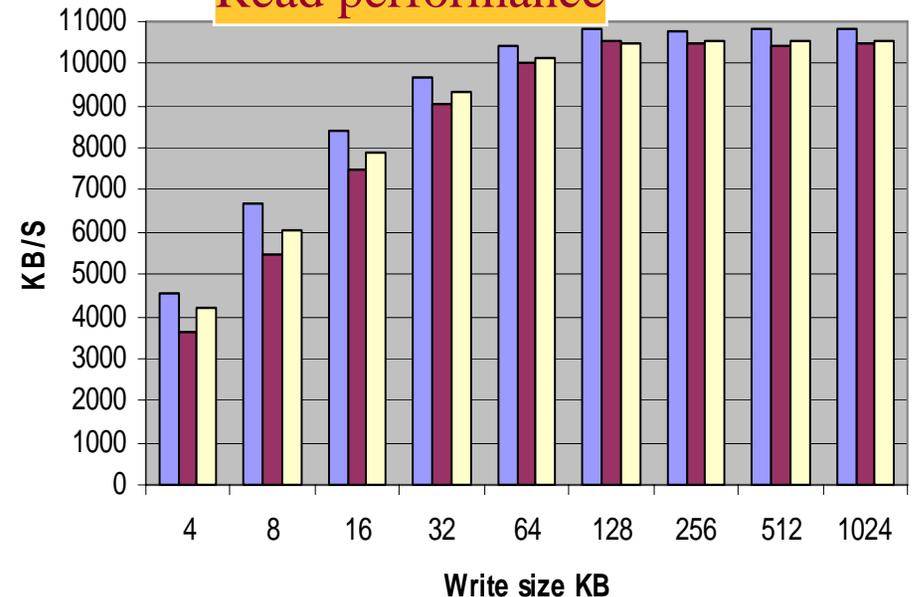
# Pipelining of Encryption



Write performance



Read performance



- Pipelining of cryptographic ops. and networking
  - Improved performance ~ 10% - 30%
- OFB ~ 7% - 3% and CBC ~ 20% - 4%
- OpenSSL CBC is inefficient