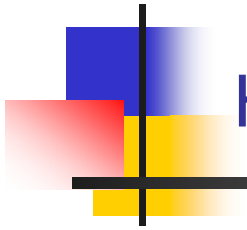
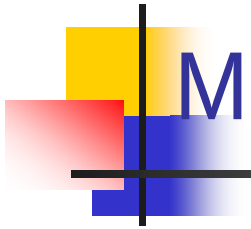


# HyFS: A Highly Available Distributed File System

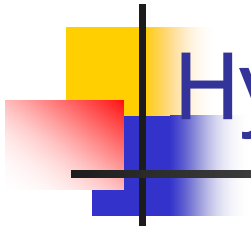
Jianqiang Luo, Mochan Shrestha, Lihao Xu
Wayne State University

FAST WIP Session
Feb. 27, 2008



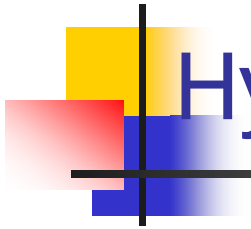
# Motivation

- High Availability Requirement
  - Storage devices are not as high available as we expect
  - In large data centers, hardware failure is a common thing
  - Proper data redundancy is the key to provide high reliability, availability and survivability
- Existing Solutions
  - Most current fault tolerance file systems use replication as the redundancy scheme, which suffers from
    - High cost to purchase hardware and later maintenance
    - Performance in writing data, multiple replication of the same data



# HyFS Idea

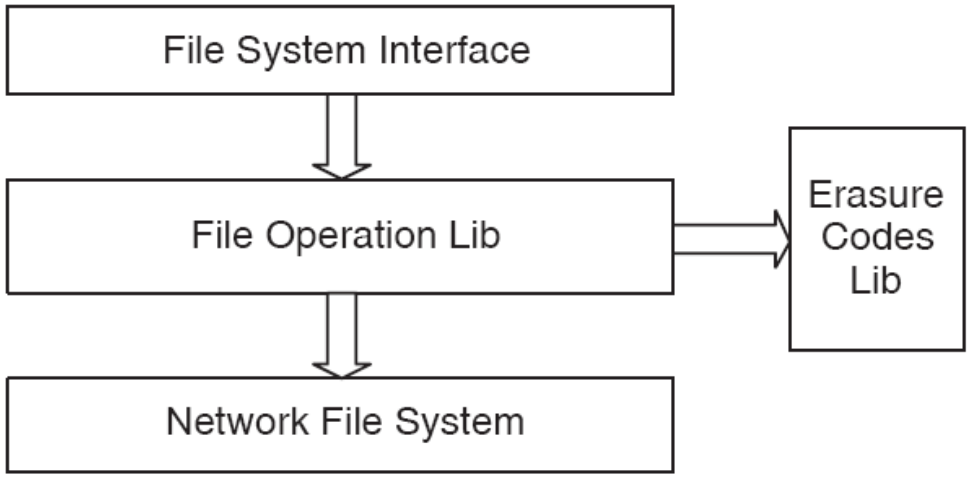
- HyFS
  - Novel use of erasure codes in file system to achieve high availability with affordable cost
    - File data will be stored to multiple storage nodes by employing erasure codes (encoding process)
    - File data can be constructed from some of these storage nodes (decoding process)
- Erasure Codes
  - MDS Erasure Codes (n, k)
    - A message has k bits, we store it as n bits by adding (n-k) redundancy bits. To recover the message, we only need to have ANY k bits among these n bits
    - Example: (n, 1) replication codes, (n, n) distribution codes, (n, n-1) parity codes
  - Erasure Codes in HyFS
    - When saving a file, we store it to n storage nodes. When reading the file, we only need to have ANY k accessible storage nodes to recover the file, thus we can tolerate up to (n-k) storage node failures



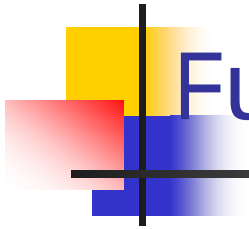
# HyFS Features

- High Flexibility
  - A general framework is designed to support *any* erasure codes to be used, and which codes is adopted depends on applications requirement
  - Easy to configure
- POSIX File API
  - POSIX file operation API is supported by a library which is independent from HyFS, and it can be readily used by applications
- File System Level
  - HyFS is a regular Linux file system, which can be installed in any popular Linux system. It has been tested on Ubuntu and SUSE
- Erasure Codes Support
  - Some erasure codes so far supported for academic research
    - Replication Code (n, 1), Distribution Code (n, n), Parity Code (n, n-1)
    - B-Code (n, n-2), X-Code (n, n-2), EVENODD Code (n, n-2), GRS Code (n, k)

# HyFS Architecture

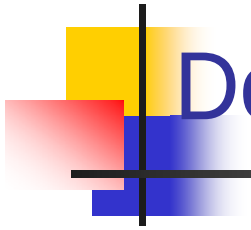



- Components
  - File System Interface (FUSE)
  - File Operation Lib (fopen, fread, fwrite, fseek, fclose)
  - Erasure Codes Lib (encode, decode)
  - Network File System (NFS)



# Future Work

- Performance Test
  - Overhead of encoding and decoding
  - Examine key factors for the HyFS performance
- Scalability
  - Study HyFS scalability when deployed to a large network
- Support More Functions
  - Latent error recovery
  - Data modification detection



# Demo

A demo at Poster Session.

Welcome to have fun with HyFS!