

# Lightweight High-performance I/O for Data-intensive Computing

**Jun Wang**

Computer Architecture and Storage System  
Laboratory (CASS)

Computer Science and Engineering Department  
University of Nebraska-Lincoln



# Background

- Technology trend
  - A steadily widening gap between CPU and disk
- Parallel I/O technique is a main weapon to address the I/O issue in HPC
  - MPI-IO
- As cluster becomes ever important in HPC, there is a need for developing scalable, high-performance parallel file system for cluster computing
  - Parallel Virtual File System and its current and future versions (PVFS2,...)
  - Lustre File System
  - IBM GPFS



# New Facts and Challenges

- Fact: developing modern high-performance parallel file system has become increasingly complex
- New Challenges
  - Temporary/derivation file I/Os are accounting for a significant percentage in many scientific and engineering applications
  - The demand imposed by the petabyte-scale data storage outstrips the ability of present underlying file storage systems
  - Intra-cluster communication becomes an ever-important issue among large-scale clusters



# Our Solution

- Develop new **portable, customized** I/O management components as **extensions** to state-of-the-art parallel file systems
- Rationale
  - Shorten the development cycle
  - Tuning to the specifics at hand could potentially increase the performance and scalability

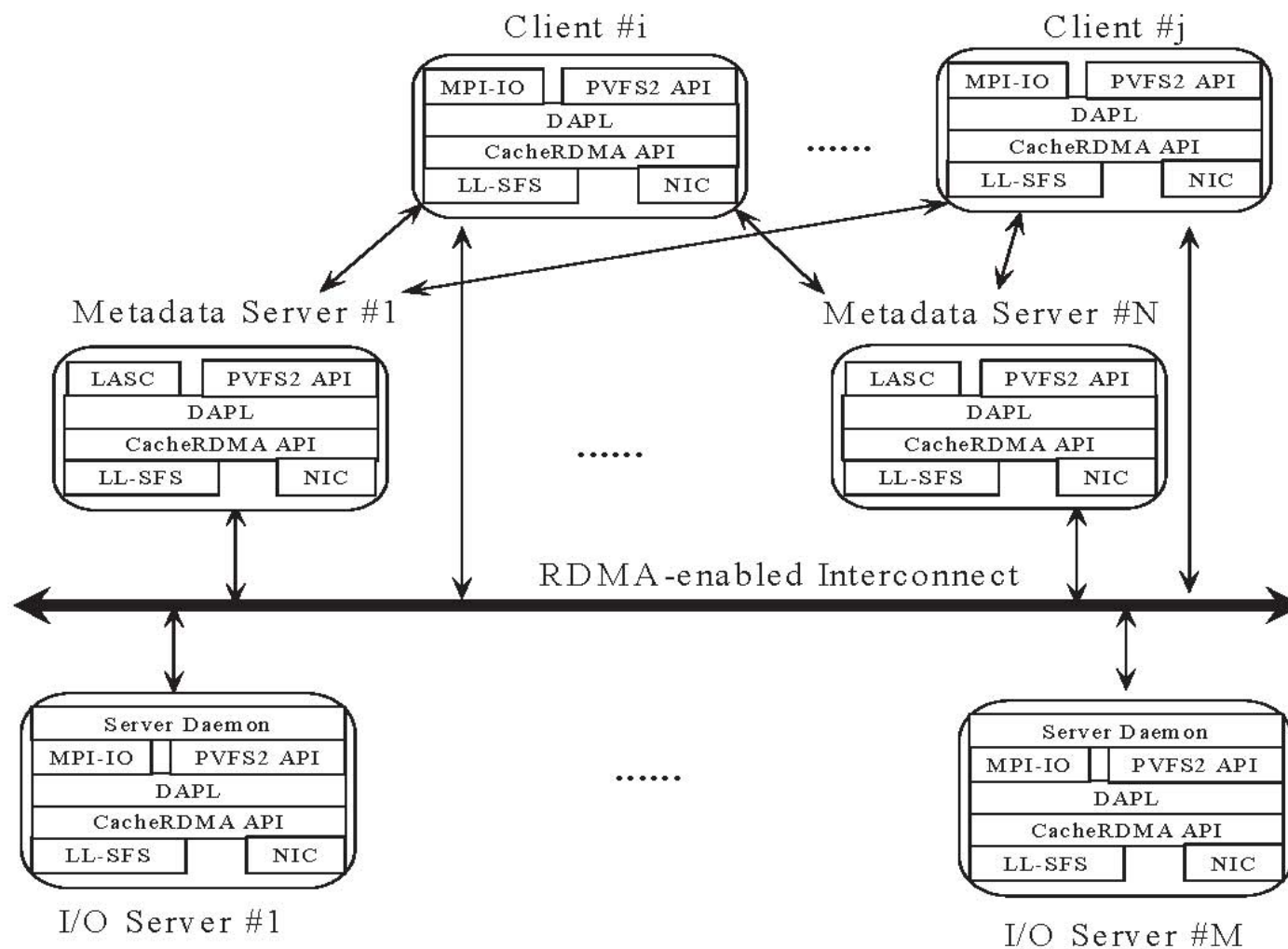


# Three Specific Research Components

- Local File I/O (User Space)
  - A lightweight, locality-aware, segment-structured local file system (LL-SFS)
- Metadata File I/O (Middleware)
  - A customized, scalable distributed file mapping scheme (Location-aware Summary Content Filters or Hierarchical Bloom filter Array)
- Intra-cluster I/O (User Space & Middleware)
  - An application-level RDMA-based I/O cache manager (CacheRDMA)

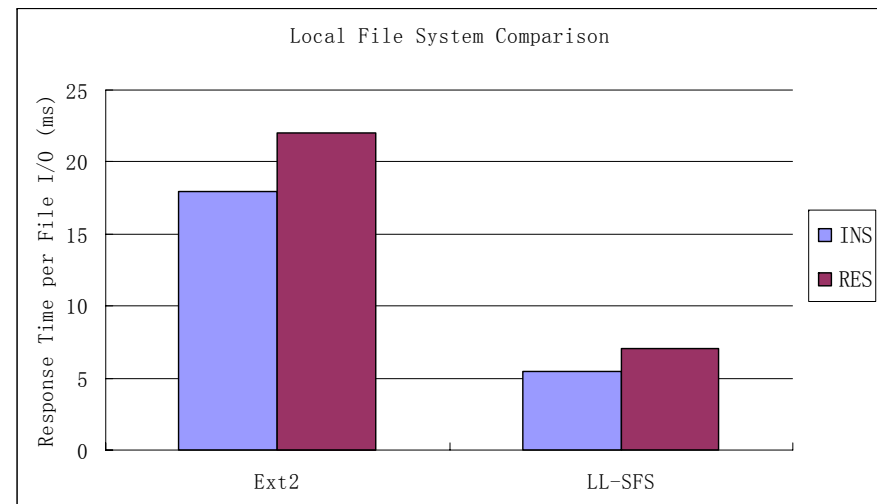


# Incorporating three lightweight I/O components into PVFS2



# Task 1: Lightweight, Local File System

- Local file I/O is critical to the overall system performance, especially for commodity PC clusters.
  - Relatively smaller buffer cache size, expensive metadata I/O, fragmented disk layout, etc.
- We are developing an LL-SFS API interface for PVFS2.
  - A specific collaborative work: Improving non-contiguous I/O performance at local file I/O level, for small file I/Os in particular.



# Task 2: A Customized, Scalable, Distributed File Mapping Scheme

- Skewed Load to Metadata
  - Metadata operations may make up to over 50% of all file system operations
- We are developing two file lookup schemes for PVFS2
  - Location-aware summary filter arrays and hierarchical bloom filter arrays
  - How to implement? E.g., work with file permission check
- We are developing novel metadata grouping schemes for multi-metadata server environment





# Task 3: Application-level RDMA-based I/O Cache Manager

- RDMA is an increasingly popular data communication technique currently adopted by cluster systems
- Study the breakdown of processing overhead for iSCSI applications over RDMA protocol suite on both client and server sides

