

# QoS-driven Storage Management for High-end Computing Systems

Ming Zhao

Computing & Information Sciences  
Florida International University  
[zhaom@cis.fiu.edu](mailto:zhaom@cis.fiu.edu)

Renato Figueiredo

Electrical & Computer Engineering  
University of Florida  
[renato@acis.ufl.edu](mailto:renato@acis.ufl.edu)

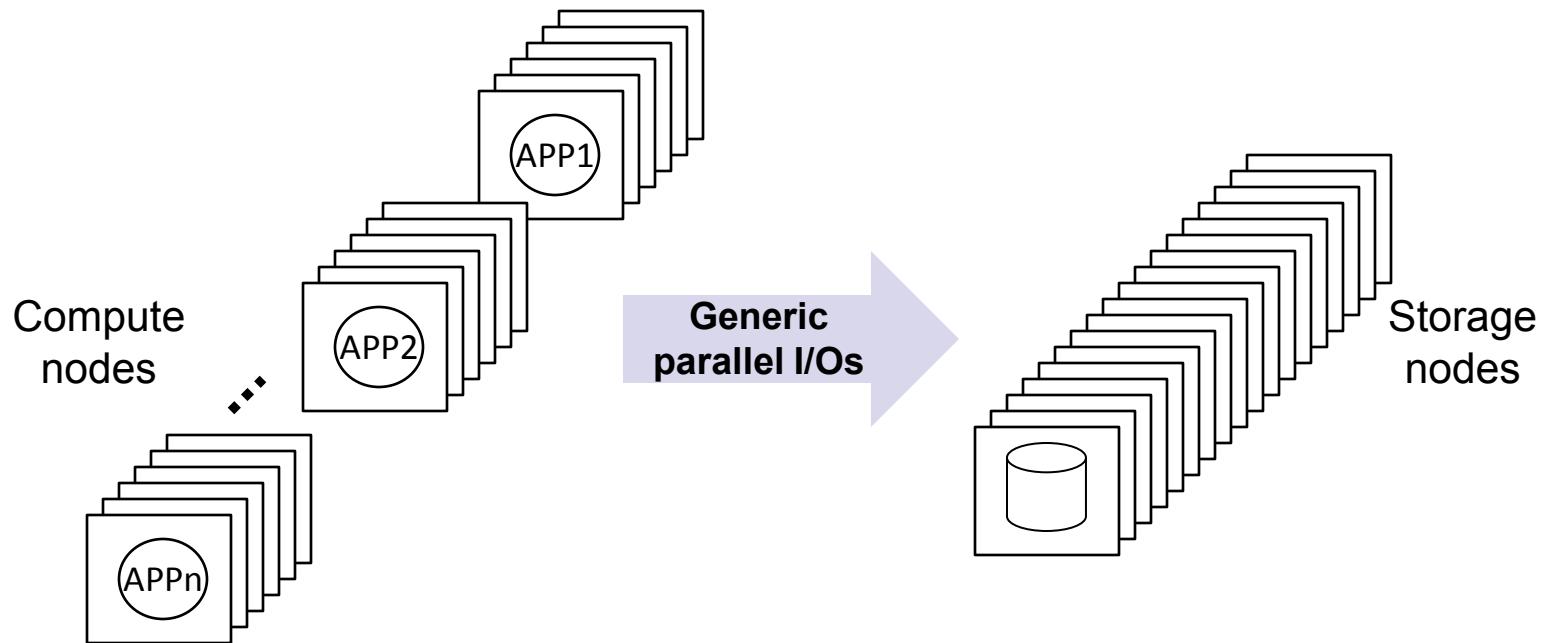


# Outline

- Motivation and objectives
- Proposed research
  - Parallel file system virtualization
  - Storage management services
  - Autonomic storage resource optimization
- Implementation plan

# Motivation

- The lack of QoS differentiation in HEC storage systems
  - Unable to recognize different application I/O workloads
  - Unable to satisfy users' different I/O performance needs



# Motivation

- The need for different I/O QoS from HEC applications
  - Diverse I/O demands and performance requirements
  - Examples:
    - WRF: Hundreds of MBs of inputs and outputs
    - mpiBLAST: GBs of input databases
    - S3D: TBs of restart files on a regular basis
- This mismatch will become even more serious in future ultra-scale HEC systems

# Research Objectives

- Per-application storage resource allocation
  - Parallel file system virtualization
- Efficient management of storage resource allocations
  - Storage management services
- Automatic optimization of storage resources usage
  - Autonomic storage resource management

# Per-application I/O Bandwidth Allocation

- Problem:
  - Lack of per-application I/O bandwidth allocation
    - Static partition of storage nodes is inflexible
    - Compute nodes based partition is insufficient
- Proposed solution:
  - Parallel file system (PFS) virtualization
    - Per-application virtual PFSs
    - Application-specific I/O bandwidth allocation per virtual PFS

# Parallel File System (PFS) Background

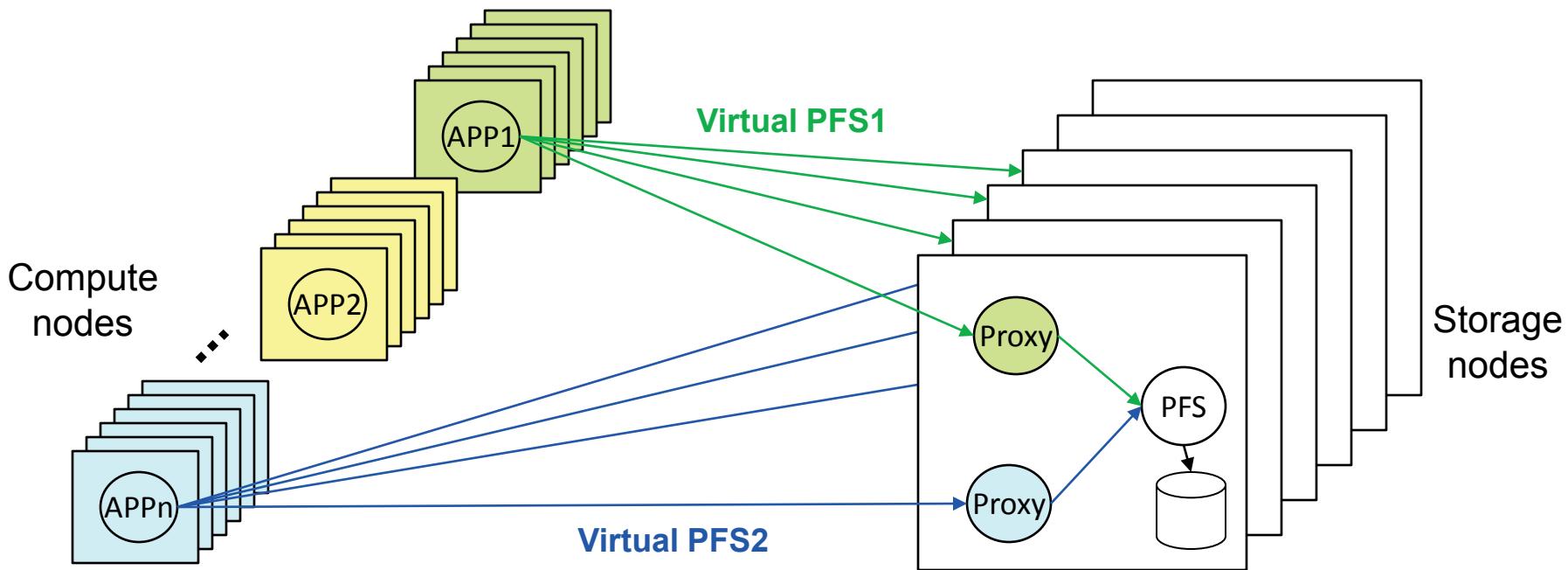
- At the core of storage resource management
  - Components: PFS clients, data servers, metadata servers
  - Examples: GPFS, IBRIX, Lustre, Panasas, PVFS etc.
- Designed for general parallel applications
  - No differentiation of different application I/Os
- Fine-tuned for overall system throughput
  - Not for specific application I/O QoS requirements

# PFS Virtualization

- Per-application virtual PFSs
  - Dynamically created and destroyed based on application lifecycles
  - Support for per-application I/O bandwidth allocation and enforcement
- Approaches:
  - Proxy based virtualization
  - PFS extension based virtualization

# PFS Virtualization

- Proxy-based PFS virtualization
  - Indirection of application I/O access
  - Creation of per-application virtual PFS



# PFS Virtualization

- Proxy based virtualization
  - Applicable to different PFS protocols
  - Seamless integration with existing HEC storage systems
  - Non-negligible overhead due to extra layer of indirection
- PFS extension based virtualization
  - Modifications on existing PFS protocols
  - Support for per-application I/O identification and handling

# Service-based Storage Management

- Problem:
  - Management of I/O bandwidth allocations for a large number of applications in a ultra-scale HEC system
- Proposed solution:
  - Service-based middleware for managing virtual PFSs
    - Storage resource scheduling
    - Storage resource monitoring

# Service-based Storage Management

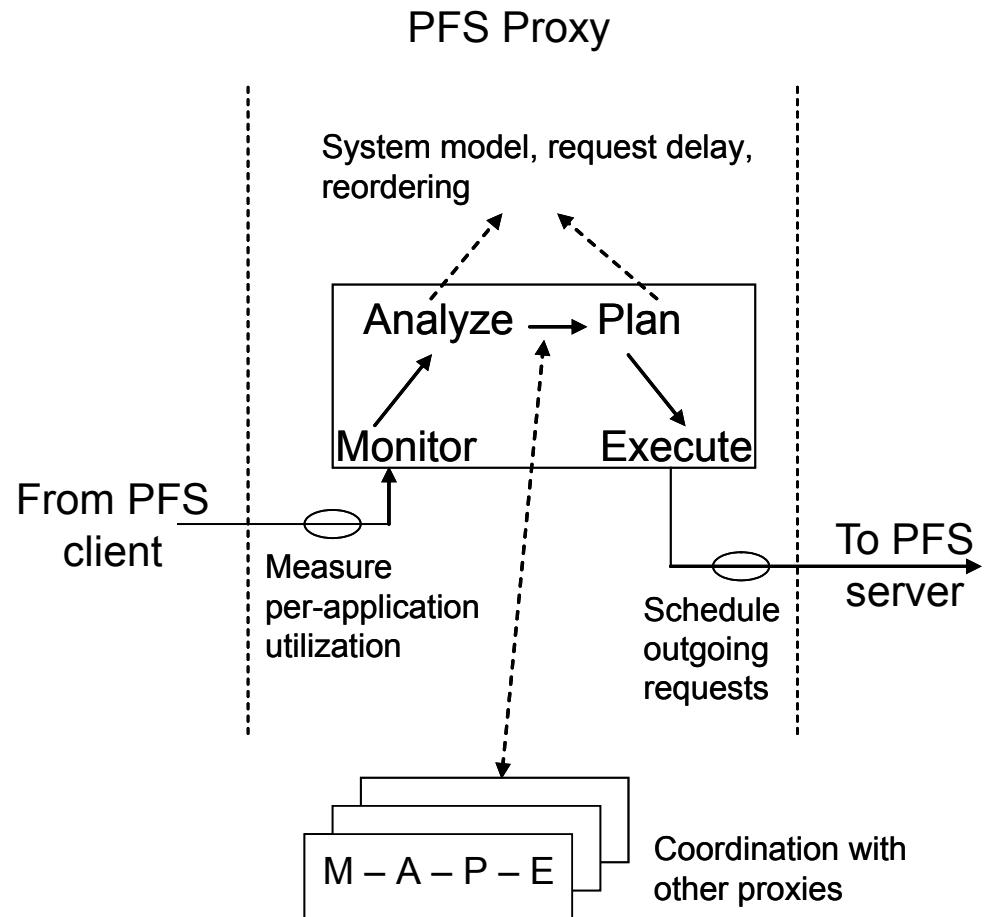
- Storage resource scheduling
  - Support for per-application reservation of I/O bandwidth
  - Integration with typical HEC job schedulers (e.g., PBS, Torque, LoadLeveler)
- Storage resource monitoring
  - Support for per-application tracking of bandwidth usage
  - Integration with typical cluster monitoring frameworks (e.g., Ganglia, ClusterMon)

# Autonomic Storage Resource Optimization

- Problem:
  - Dynamic resource scheduling for fair sharing of storage resources
  - Automatic optimization of I/O bandwidth utilization
- Proposed research:
  - Autonomic storage resource management upon the virtualized PFS infrastructure

# Autonomic Storage Resource Optimization

- Proxy-based autonomic I/O control loop
- Dynamic scheduling algorithms (e.g., SFQ)
- Optimization based on coordinated scheduling



# Implementation Plan

- Prototyping
  - Leverage typical open-source PFSs (e.g., PVFS, Lustre)
  - Consider commercial PFSs through collaboration (e.g., GPFS, Panasas)
    - Seetharami Seelam, IBM T.J. Waston
- Evaluation
  - Testbed based on in-house resources and emulation
  - Leverage real HEC site traces and resources through collaboration

# Education Plan

- Virtual machine based educational modules
  - Parallel storage systems prepackaged in virtual appliances
  - UF Grid Appliance project
    - Peer-to-peer networked virtual machine based grid computing
- Engagement of underrepresented students
  - FIU — one of the nation's largest minority-serving research institutes
    - Computer Science — 15% of the nation's Hispanic Ph.D. students

# More Information

- Ming Zhao
  - <http://www.cis.fiu.edu/~zhaom>
  - zhaom@cis.fiu.edu
- Renato Figueiredo
  - <http://www.acis.ufl.edu/~renato>
  - renato@acis.ufl.edu
- *Questions?*