

# Global File Systems and Logistical Networking

*Micah Beck*  
*Associate Professor*

Logistical Computing and Internetworking Lab  
Computer Science Department  
University of Tennessee

NASA/IEEE MSST College Park

18 May 06



**LOCI**

LOGISTICAL COMPUTING AND  
INTERNETWORKING LAB



UNIVERSITY OF TENNESSEE

# Conventional File Systems

- » File descriptors
  - Define a mapping from logical file extent to physical storage resources
  - Physical resources managed by a supervisor
- » Directories
  - Define a mapping from logical namespace to file descriptors
  - File descriptors managed by a supervisor
- » Operations on directories & files
  - Defined in fixed service APIs
  - Implemented by fixed supervisory code

# A Design Principle

- » In a layered service architecture, any function should be implemented at the highest layer at which it can be implemented completely and correctly
  - Otherwise, design will not meet evolving needs of a broad user community
  - Otherwise, design cannot incorporate new underlying technologies
- » Examples
  - Multics, Unix, RISC, RAID, IP

# Problems with Global/Parallel File Systems

- » Standard APIs require close synchronization in the management of directory and file descriptor data structures
- » Supervisory managers create bottlenecks
- » Application-adapted operations must be implemented through complex adaptive APIs
  - Striping
  - Block size
  - Cache behavior
- » Common administrative functions require close coordination of operations

# Applying the Design Principle

- » Expose physical storage resources to the user/application layer
  - Internet Backplane Protocol
- » Implement file descriptors at user/application layer, operations adapted as necessary
  - exNode data structure for logical-to-physical mapping
  - Logistical Runtime System implements API
- » Synchronization and coordination due to shared resources and administrative mechanisms can be adapted to requirements, minimized

# Sample exNodes

