Mirror File System

A Multiple Server File System

John Wong
CTO
Twin Peaks Software Inc.
John.Wong@TwinPeakSoft.com

Multiple Server File System

- Conventional File System -- UFS, EXT3 and NFS
  - Manage and store files on a single server and its storage devices
- Multiple Server File system
  - Manage and store files on multiple servers and their storage devices

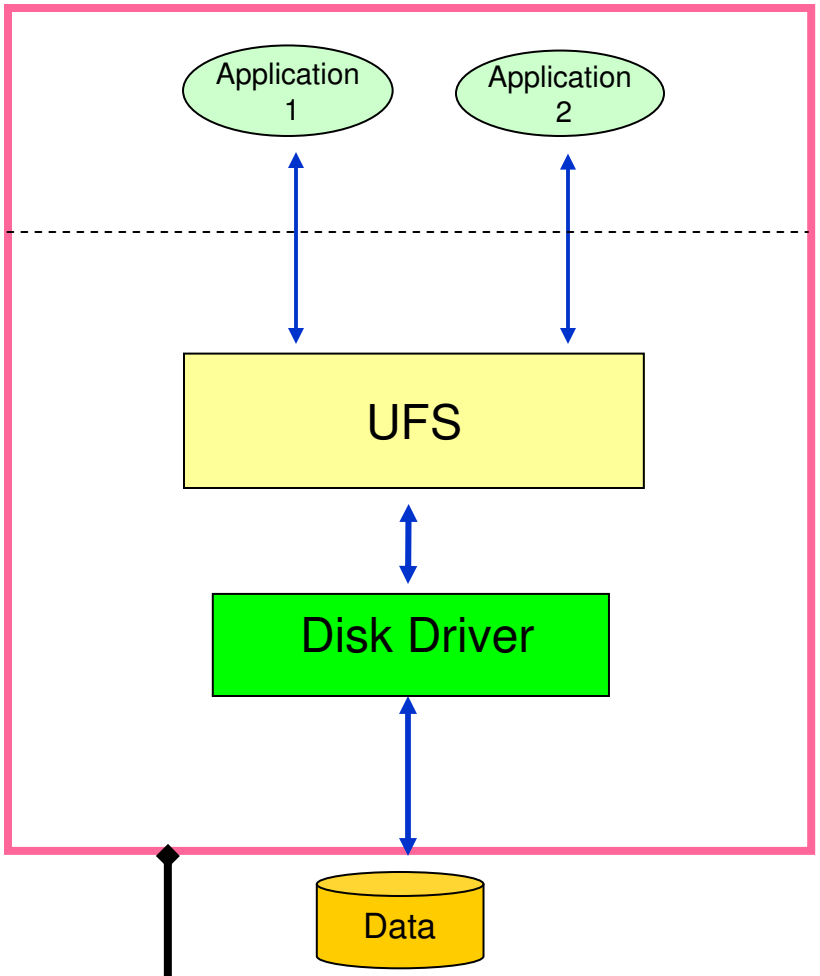
Problems

- Single resource is vulnerable
- Redundancy provides a safety net
  - Disk level => RAID
  - Storage level => Storage Replication
  - TCP/IP level => SNDR
  - File System level => CFS, MFS
  - System level => Clustering systems

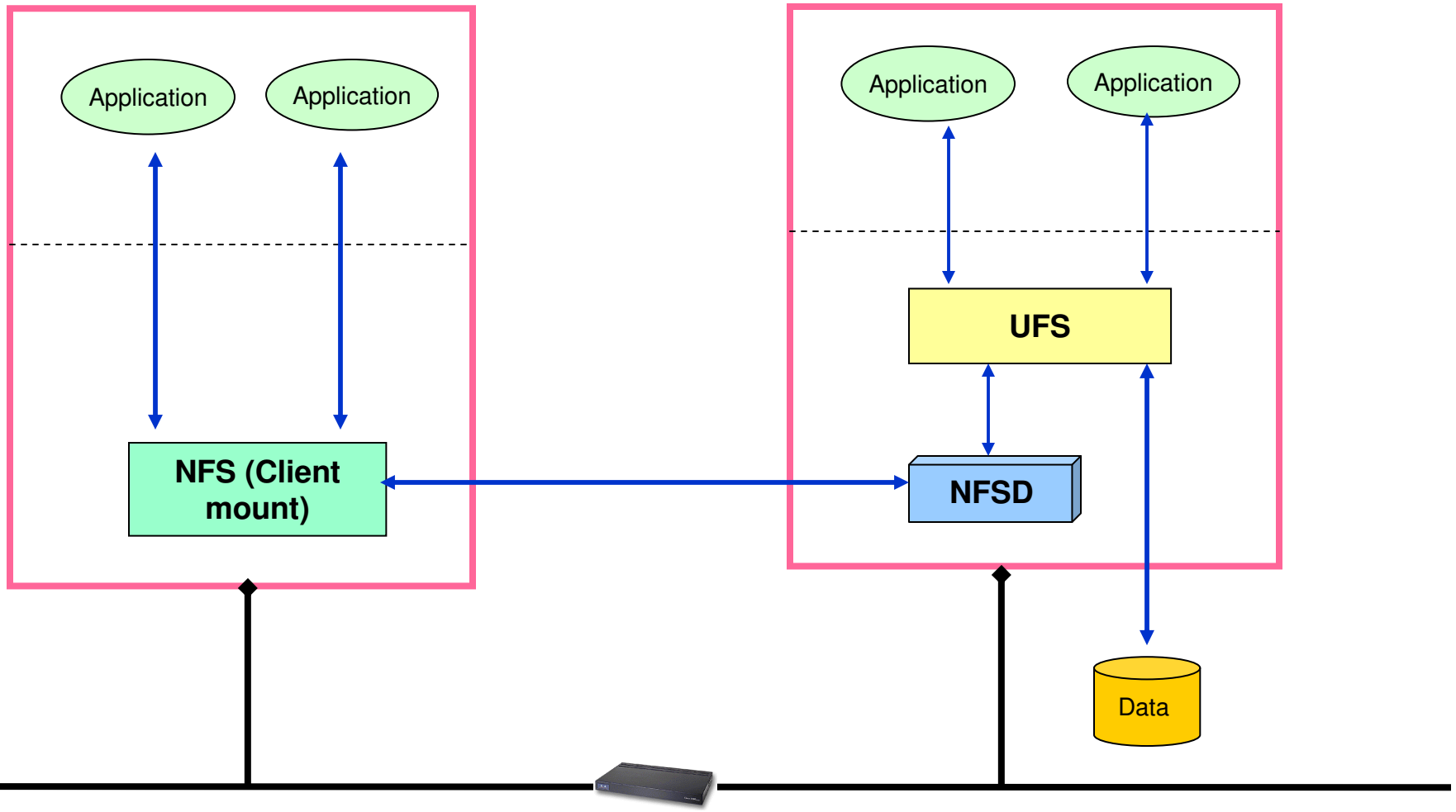
Why MFS?

- Better Disaster Recovery
- Better RAS
- Better Scalability
- Better Performance
- Better Resources Utilization

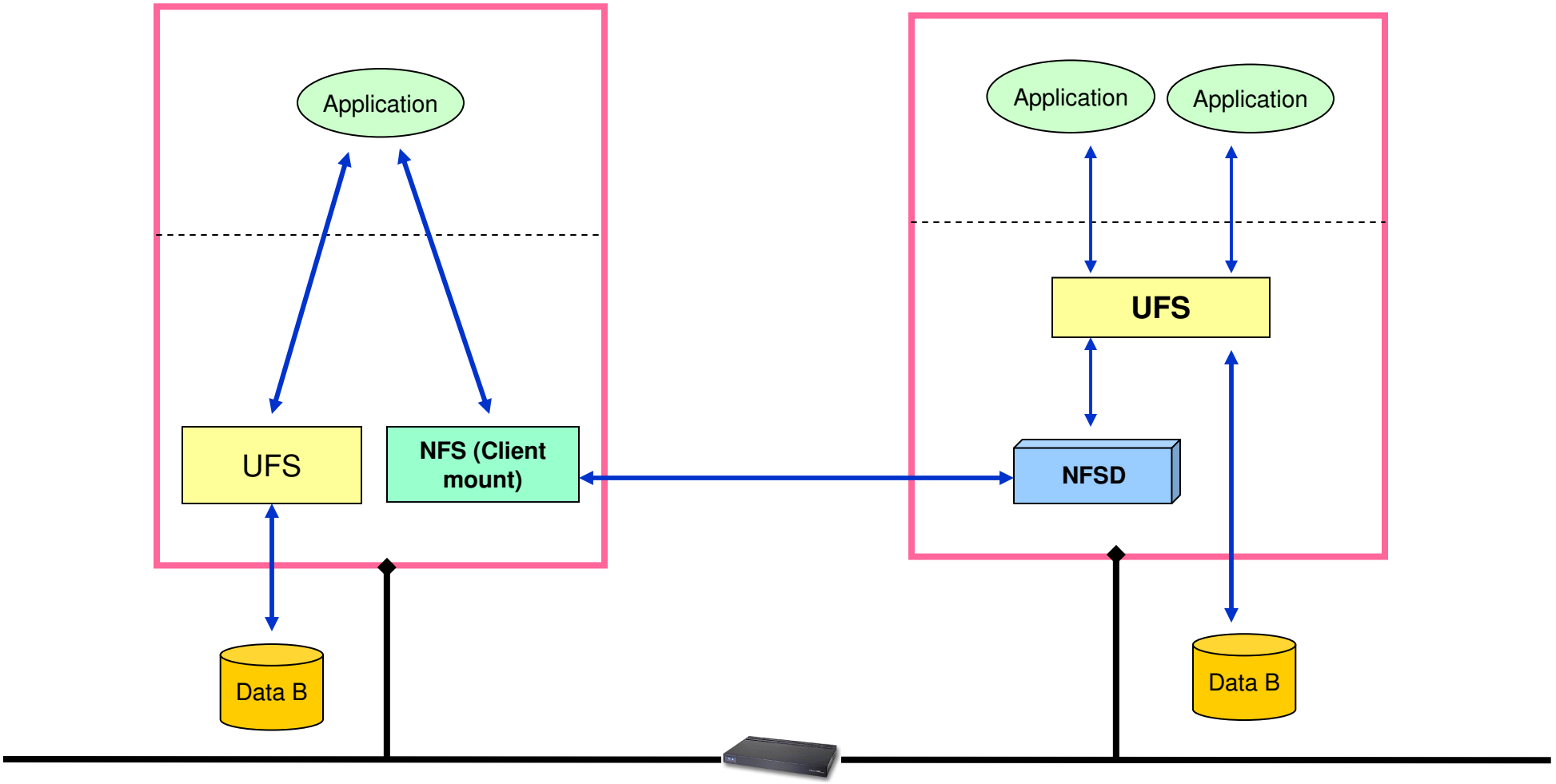
Unix File System



Network File System



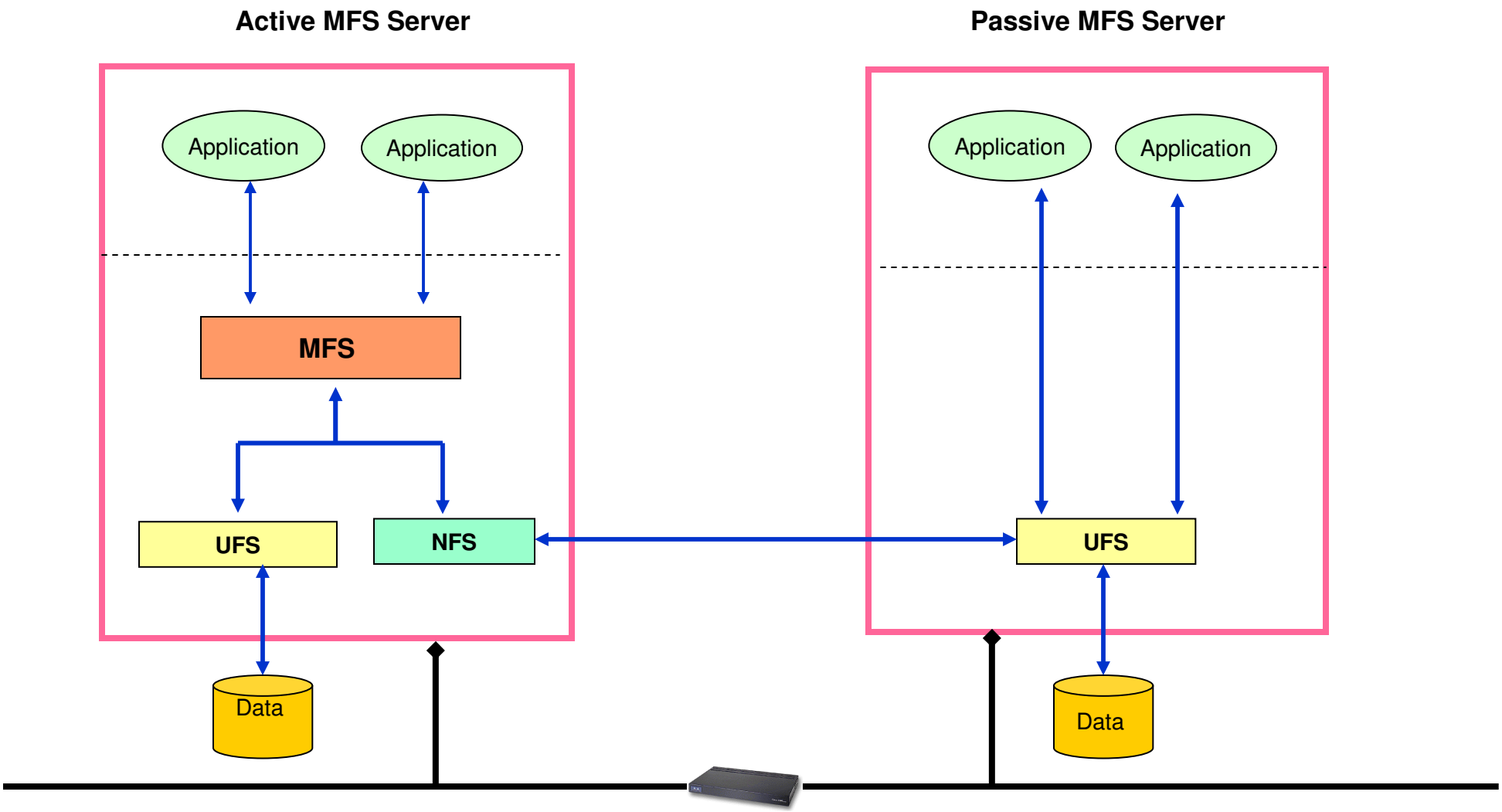
UFS | NFS


Application
UFS
NFS (Client mount)
Data B
Application
Application
UFS
NFSD
Data B

UFS + NFS

- UFS manages data on the local server's storage devices
- NFS manages data on remote server's storage devices
- Combine these two file systems to manage data on both local and remote servers storage devices

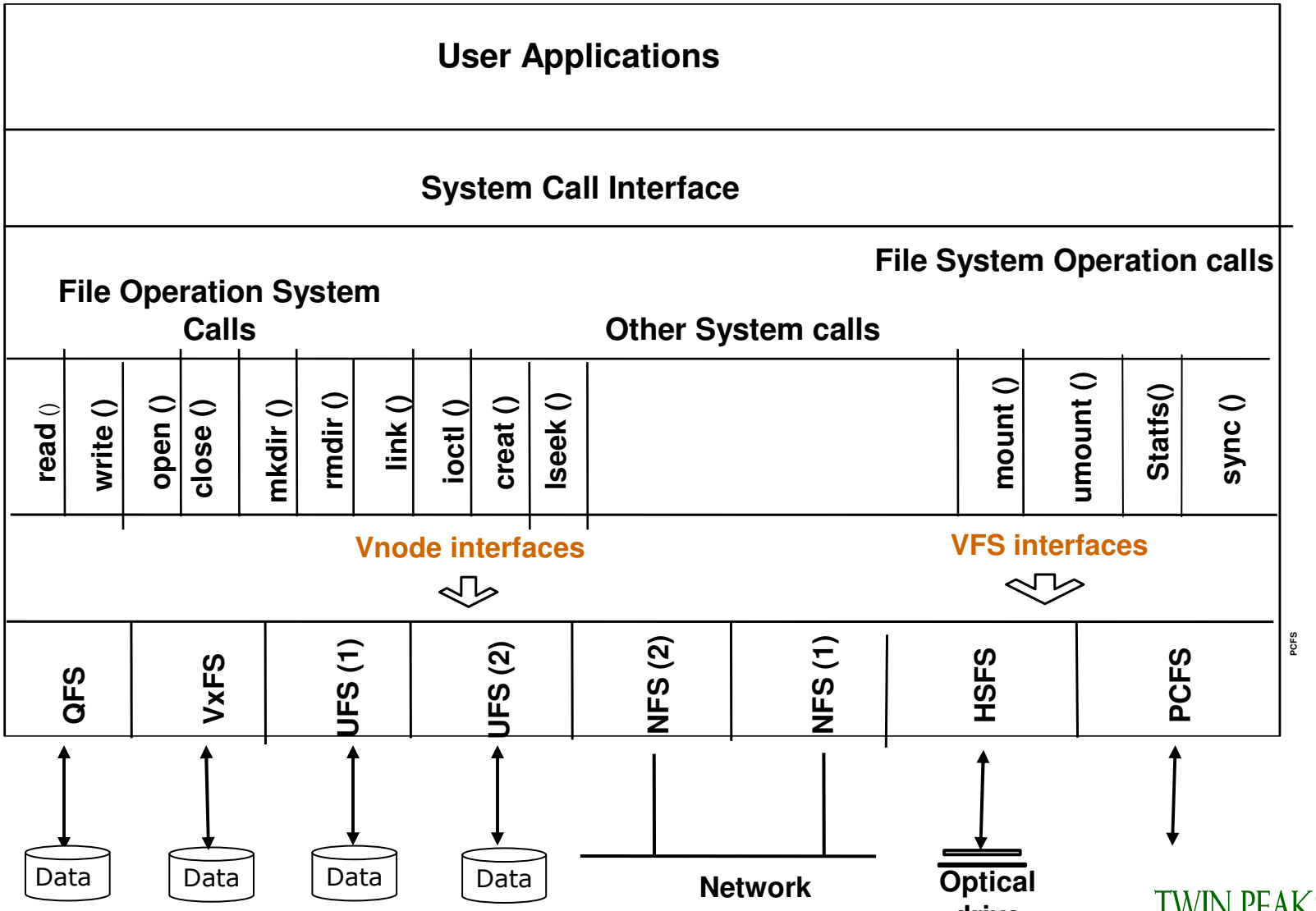
MFS = UFS + NFS


Active MFS Server
Passive MFS Server
Application
Application
Application
Application
MFS
UFS
NFS
UFS
Data
Data

Building Block Approach

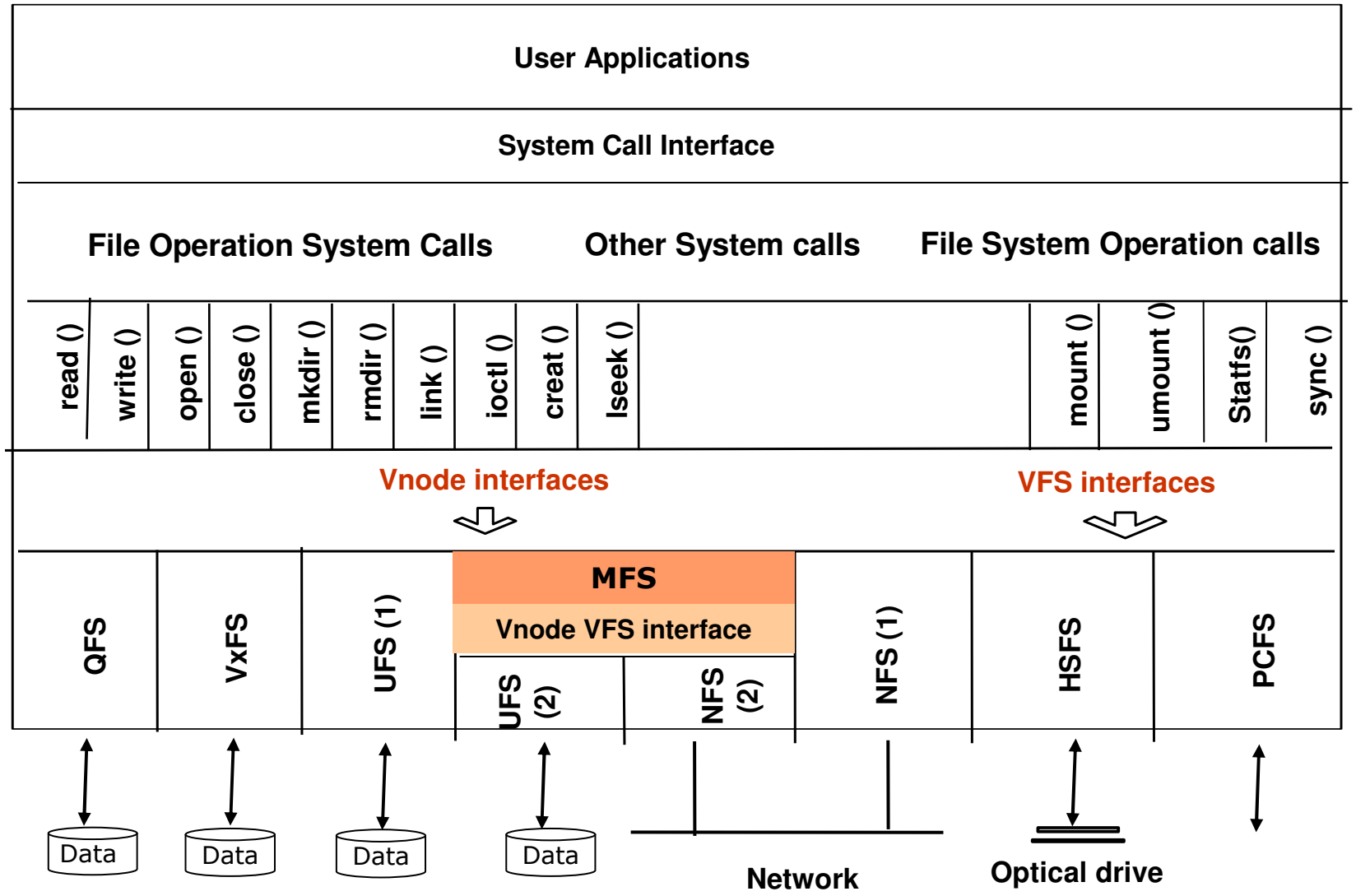
- MFS is a kernel loadable module
- MFS is loaded on top of UFS and NFS
- Standard VFS interface
- No change to UFS and NFS

File System Framework



SOLARIS Internal, Core Kernel Architecture, Jim Mauro. Richard McDougall, PRENTICE HALL

MFS Framework



Transparency

- Transparent to users and applications
  - No re-compilation or re-link needed
- Transparent to existing file structures
  - Same pathname access
- Transparent to underlying file systems
  - UFS, NFS

Mount Mechanism

- Conventional Mount
  - One directory, one file system
- MFS Mount
  - One directory, two or more file systems

Mount Mechanism

mount –F mfs host:/ndir1/ndir2 /udir1/udir2

- First mount the NFS on a UFS directory
- Then mount the MFS on top of UFS and NFS
- Existing UFS tree structure /udir1/udir2 becomes a local copy of MFS
- Newly mounted host:/ndir1/ndir2 becomes a remote copy of MFS
- Same mount options as NFS except new ‘-o nolock’ option and no ‘-o hard’ option

READ/WRITE Vnode Operation

- All VFS/vnode operations received by MFS
- READ related operation: read, getattr,….
  those operation only need to go to local copy (UFS).
- WRITE related operation: write, setattr,…..
  those operations go to both local (UFS) and remote (NFS) copy simultaneously (using threads)

Mirroring Granularity

- Directory Level
  - Mirror any UFS directory instead of entire UFS file system
  - Directory A mirrored to Server A
  - Directory B mirrored to Server B
- Block Level Update
  - Only changed block is mirrored

MFS mfsck Command

/usr/lib/fs/mfs/mfsck mfs_dir

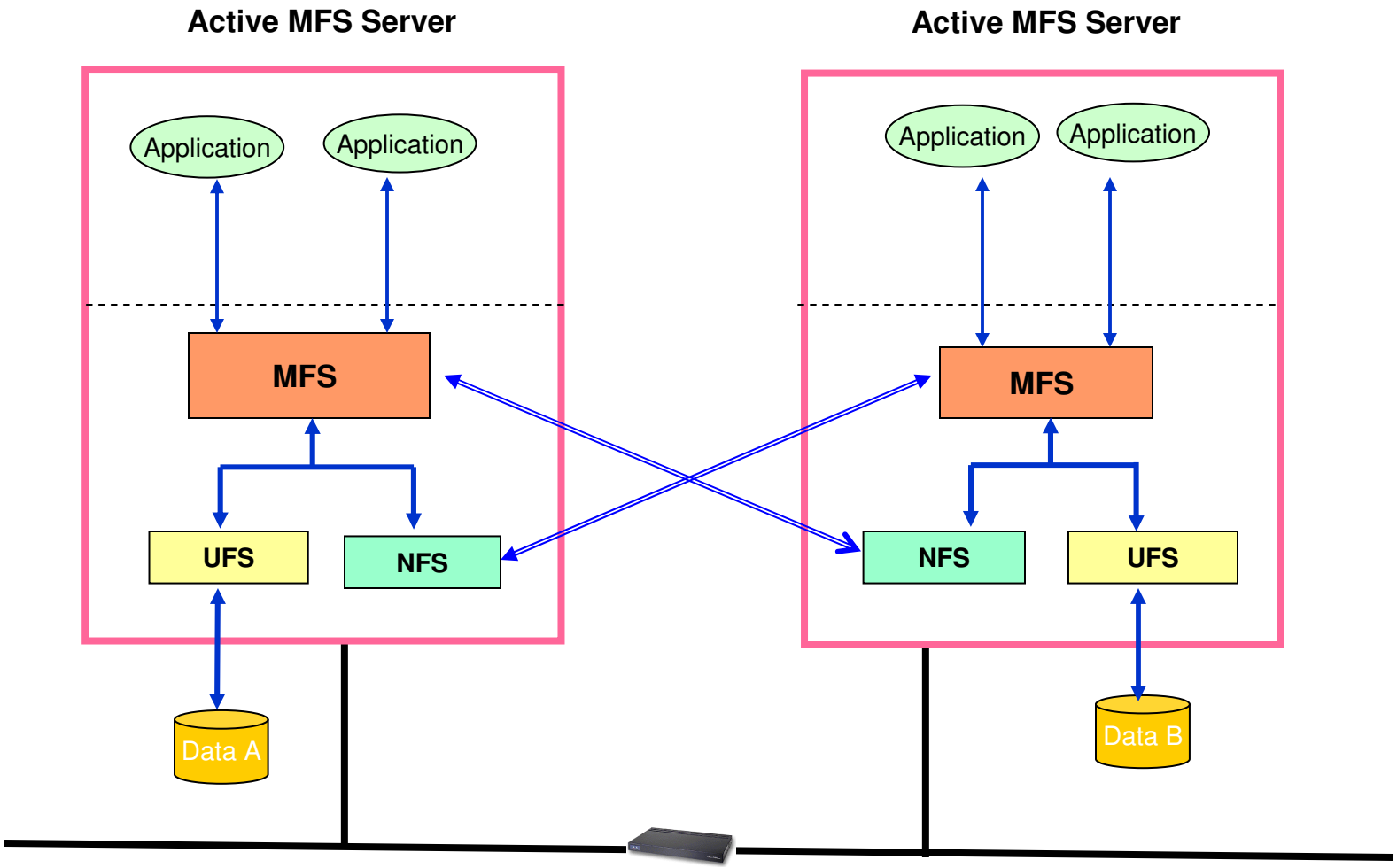
- After MFS mount succeeds, the local copy may not be identical to the remote copy.
- Use mfsck (the MFS fsck) to synchronize them.
- The mfs_dir can be any directory under MFS mount point.
- Multiple mfsck commands can be invoked at the same time.

MFS msync Command

/usr/lib/fs/mfs/msync mfs_root_dir

- A daemon that synchronizes MFS pair after a remote MFS partner fails.
- Upon a write failure, MFS:
  - Logs name of file to which the write operation failed
  - Starts a heartbeat thread to verify the remote MFS server is back online
- Once the remote MFS server is back online, msync uses the log to sync missing files to remote server.

Active/Active Configuration



MFS Locking Mechanism

MFS uses UFS, NFS file record lock.

Locking is required for the active-active configuration.

Locking enables write-related vnode operations as atomic operations.

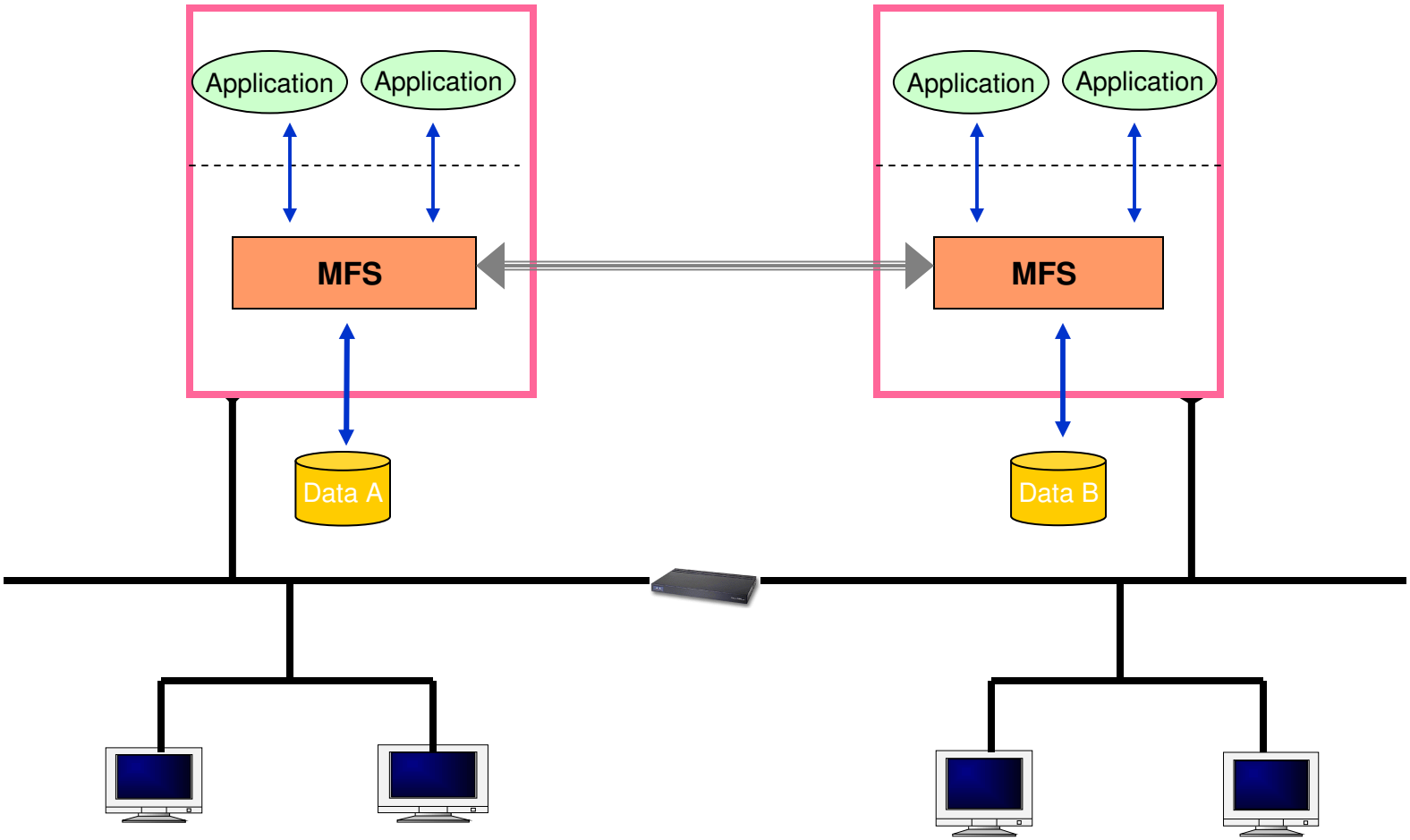
Locking is enabled by default.

Locking is not necessary in active-passive configuration.

Summary

- Multiple locations of real-time, active data
- Full utilization of hardware/software investment
  - no expensive “spare tires”
- An efficient DR solution
  - no cumbersome recovery procedures
- Greatly reduced network traffic
  - go to your nearest server
- Scalable, pay as you grow
- Easy to deploy and manage

MFS


Application
Application
Application
Application
MFS
MFS
Data A
Data B