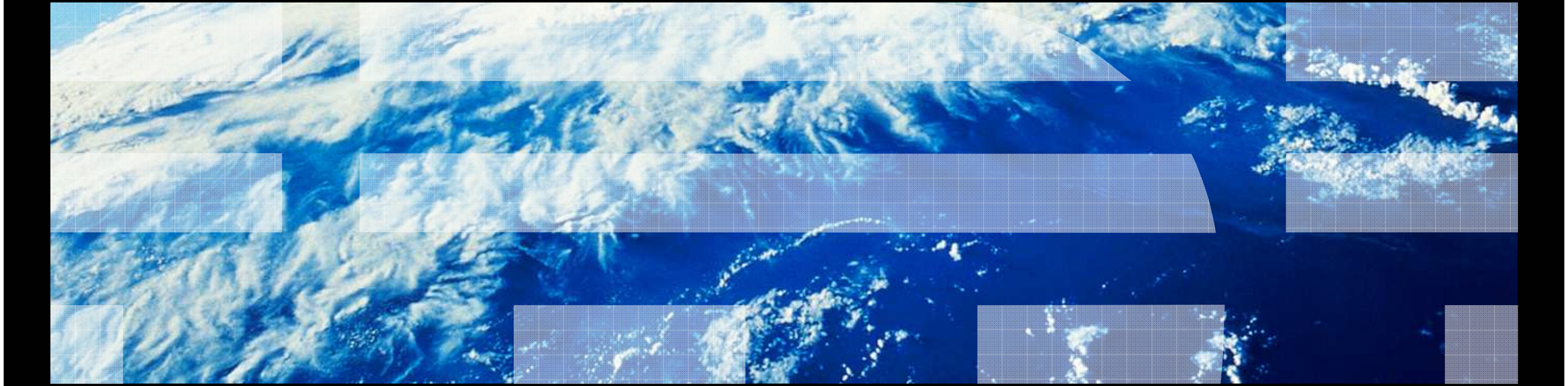
Dean Hildebrand – Research Staff Member

PDSW 2009

pNFS, POSIX, and MPI-IO: A Tale of Three Semantics

Dean Hildebrand, Roger Haskin — IBM Almaden
Arifa Nisar — Northwestern University



Agenda

Motivation

pNFS

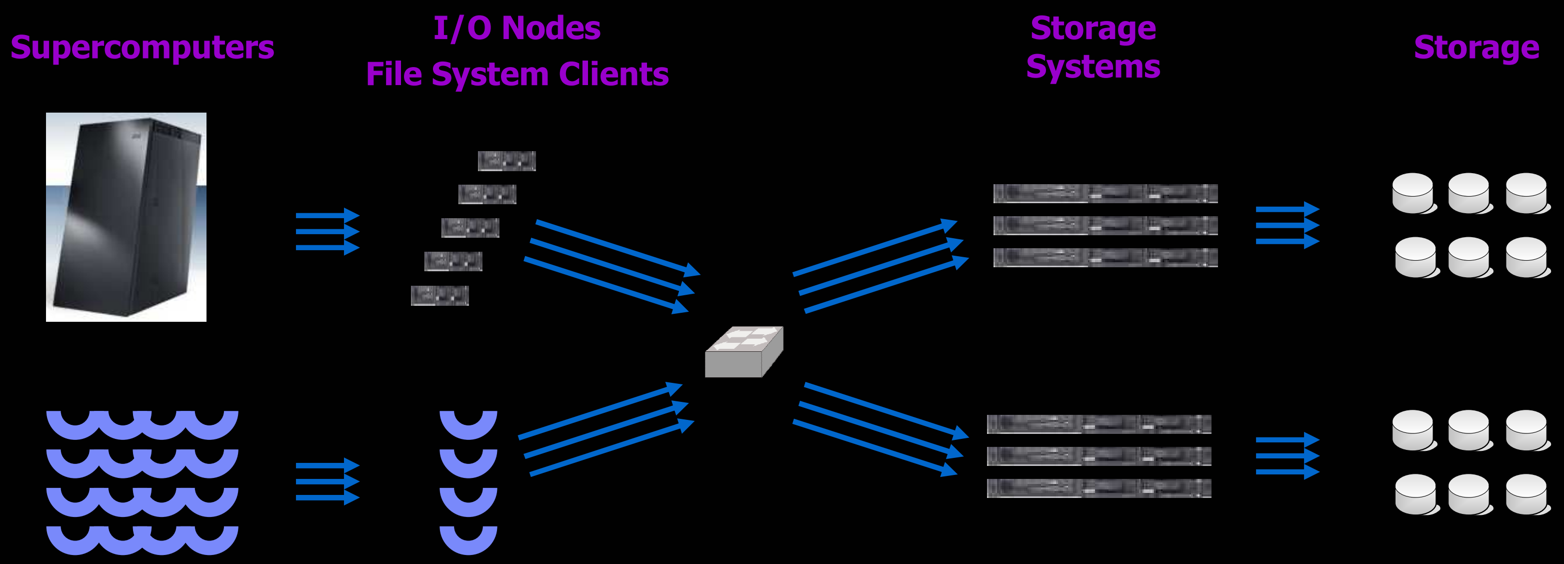
HPC consistency requirements

Protocol consistency semantics

NFSv3 ADIO driver

pNFS ADIO driver

Motivation: Commodity parallel file system clients



- Supercomputers can be connected with multiple parallel file systems
  - GPFS, PVFS2, PanFS, Lustre
- Want single file system client to access all available storage systems

Agenda

Motivation

pNFS

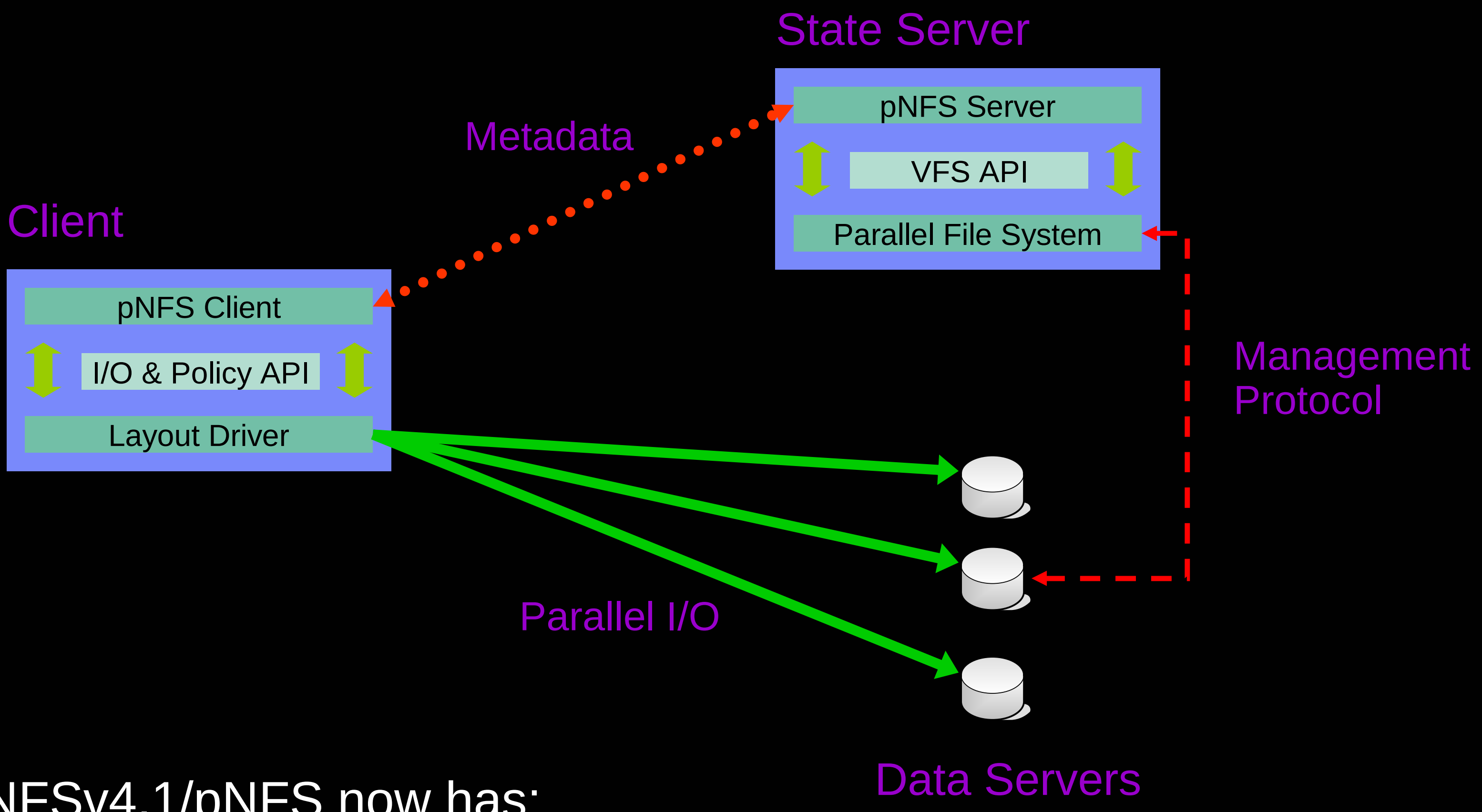
HPC consistency requirements

Protocol consistency semantics

NFSv3 ADIO driver

pNFS ADIO driver

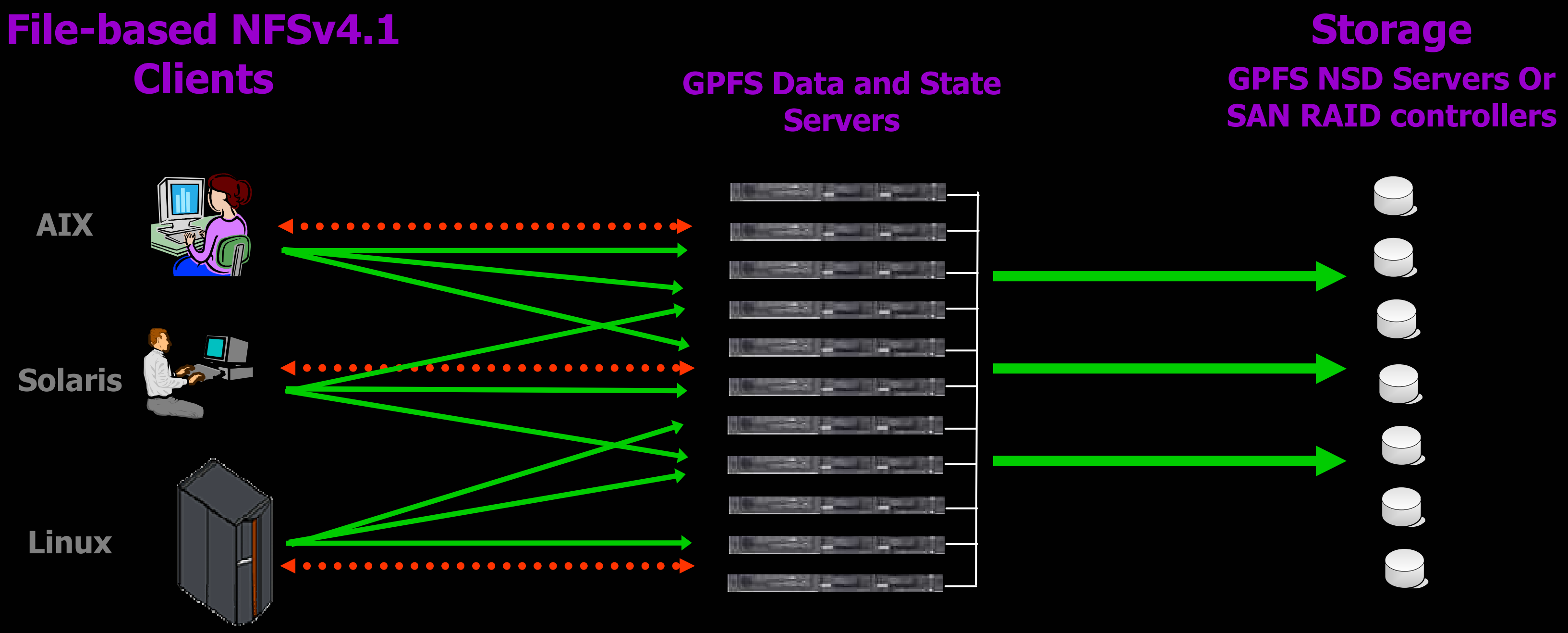
pNFS



NFSv4.1/pNFS now has:

- Integrated locking protocol
- Open/Close with per-object change attribute

pNFS with GPFS


File-based NFSv4.1 Clients
GPFS Data and State Servers
Storage
GPFS NSD Servers Or SAN RAID controllers
AIX
Solaris
Linux

- Fully-symmetric GPFS architecture - scalable data *and metadata*
  - pNFS client can mount and retrieve layout from any GPFS node
  - metadata requests can be load balanced across cluster
- pNFS server and native GPFS clients can share the same file system
  - Backup, dedup, and other mgmt functions don't need to be done over NFS
- Need robust interface between NFSD and GPFS

Agenda

Motivation

pNFS

HPC consistency requirements

Protocol consistency semantics

NFSv3 ADIO driver

pNFS ADIO driver

HPC consistency requirements

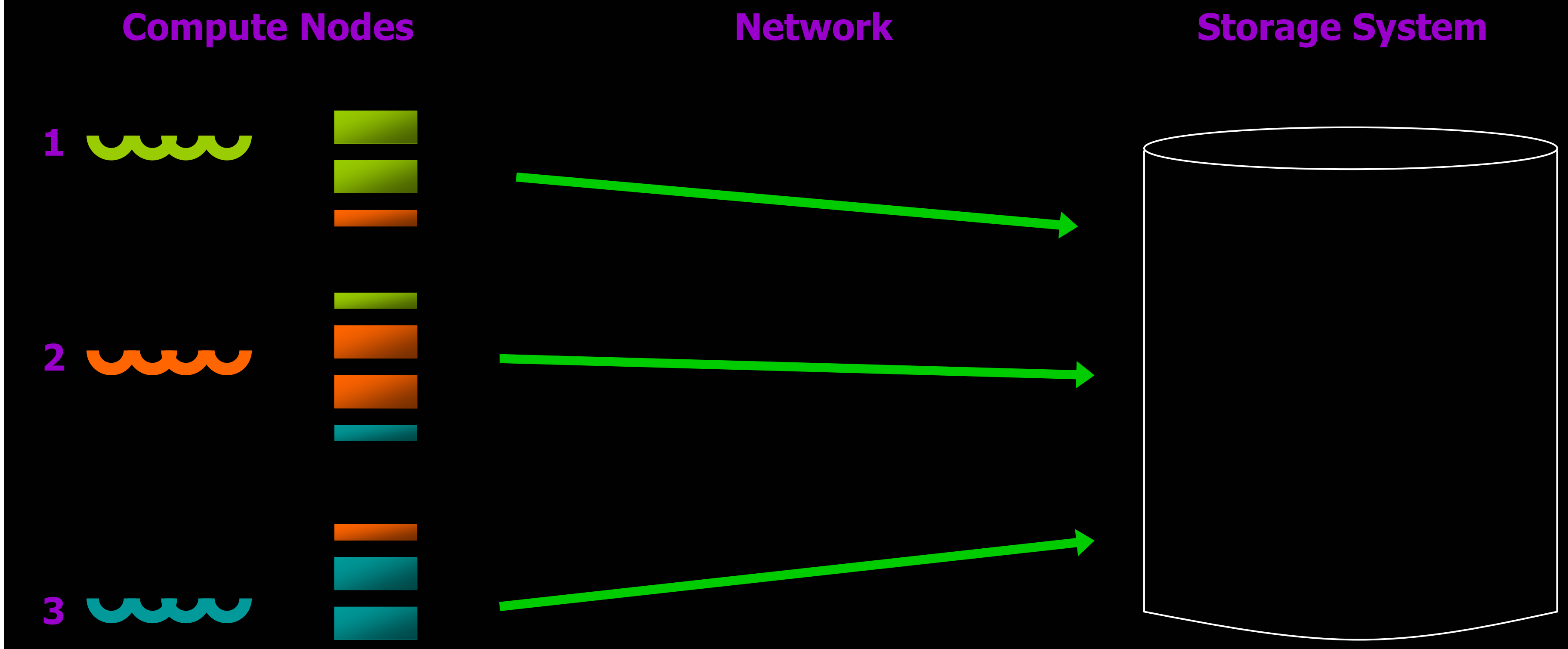
- Different I/O workloads have different requirements
  - Checkpoint
    - Write-only
    - No revalidation to new file
  - Ingest/Restart
    - Read-only
    - Revalidation on Open
  - sync-barrier-sync
    - Sync data to disk
    - Force revalidate/invalidate
  - Others?

MPI-IO: sync-barrier-sync

- Applications sometimes need to share computed results among processes/nodes
- Want to avoid MPI atomic mode to improve performance
  - Requires enforcing strict consistency semantics
  - Writes must be immediately visible by other processes
- Allows compute nodes to synchronize I/O operations between themselves
  - Sync #1 guarantees that the data written by all nodes is transferred to storage.
  - Barrier ensures that writes on all nodes complete prior to reads.
  - Sync #2 guarantees that all transferred data is visible to all processes.

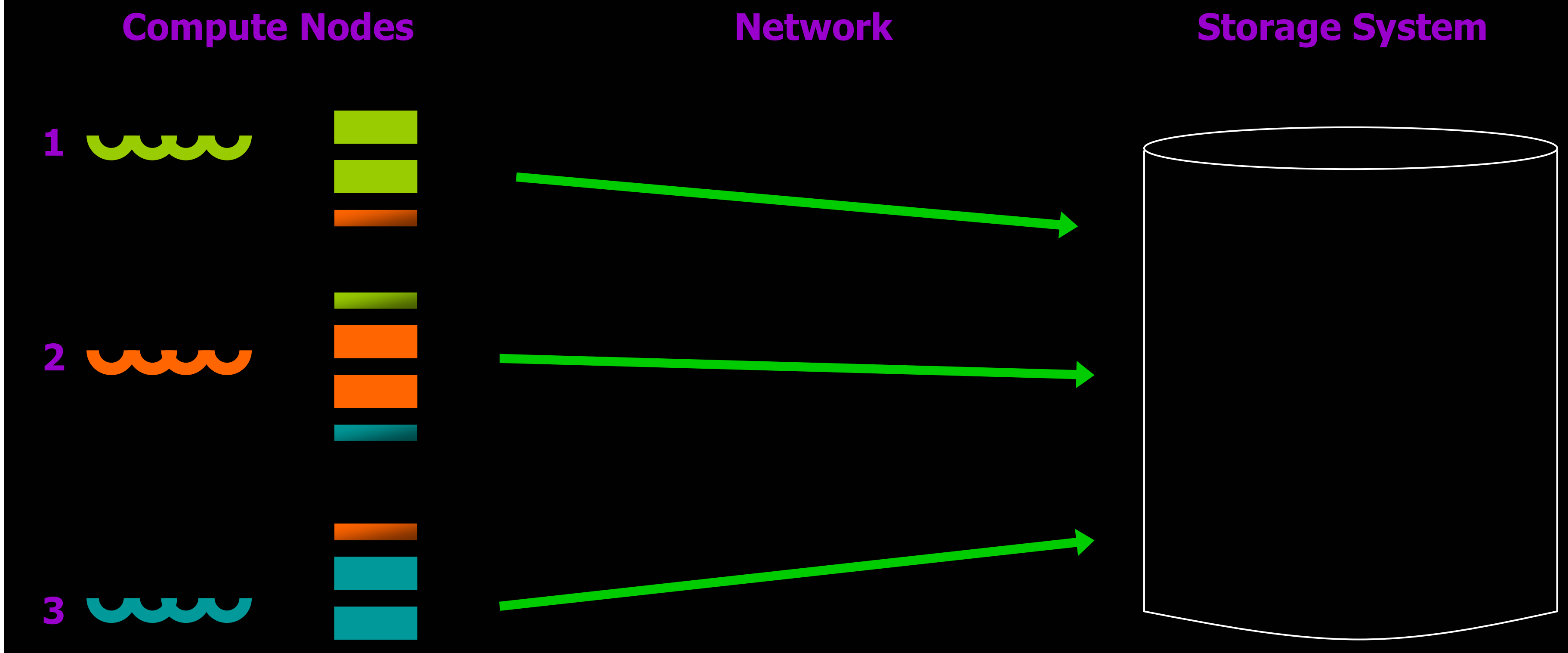
Sync-Barrier-Sync

Each node has data in its cache



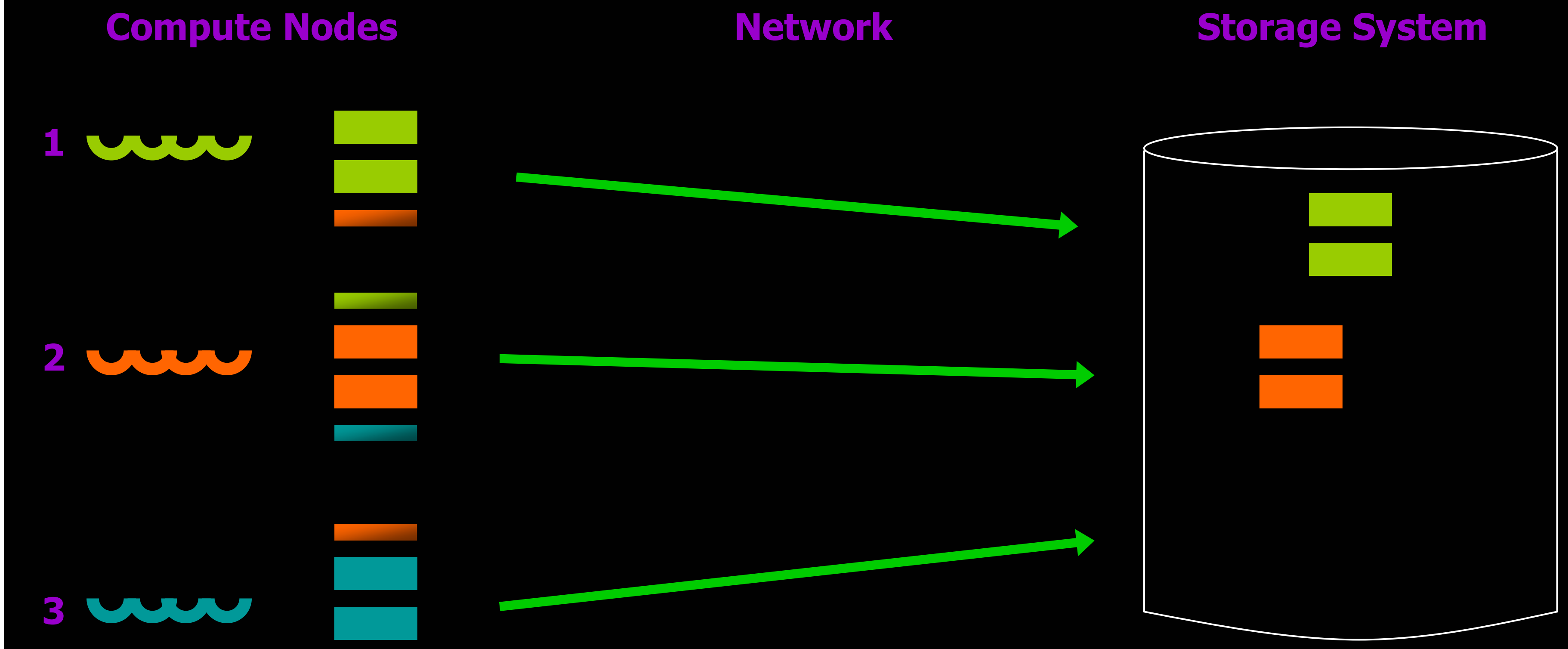
Sync-Barrier-Sync

1. SYNC: Place written data on filing servers



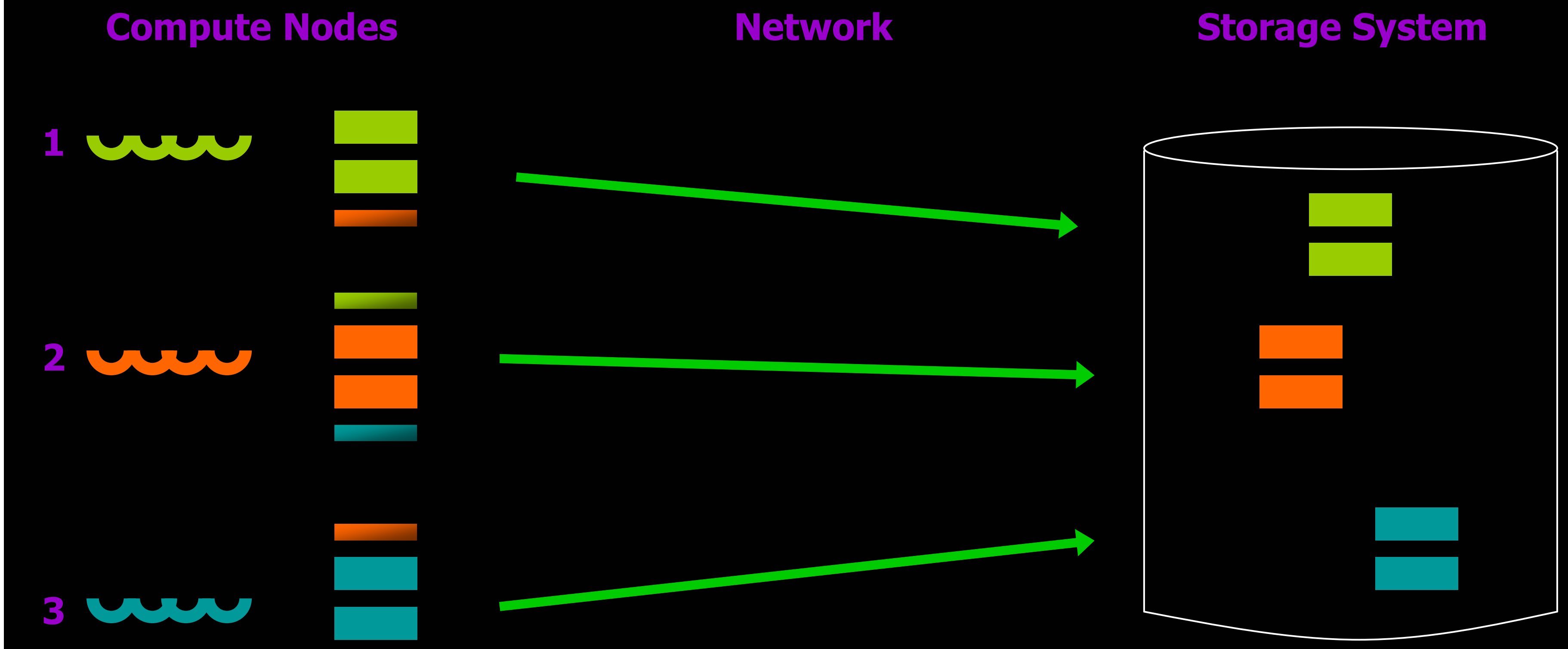
Sync-Barrier-Sync

2. BARRIER: Clients wait for other clients to flush dirty data to the servers, ensuring that no client issues read requests until all clients have the same view of file contents.



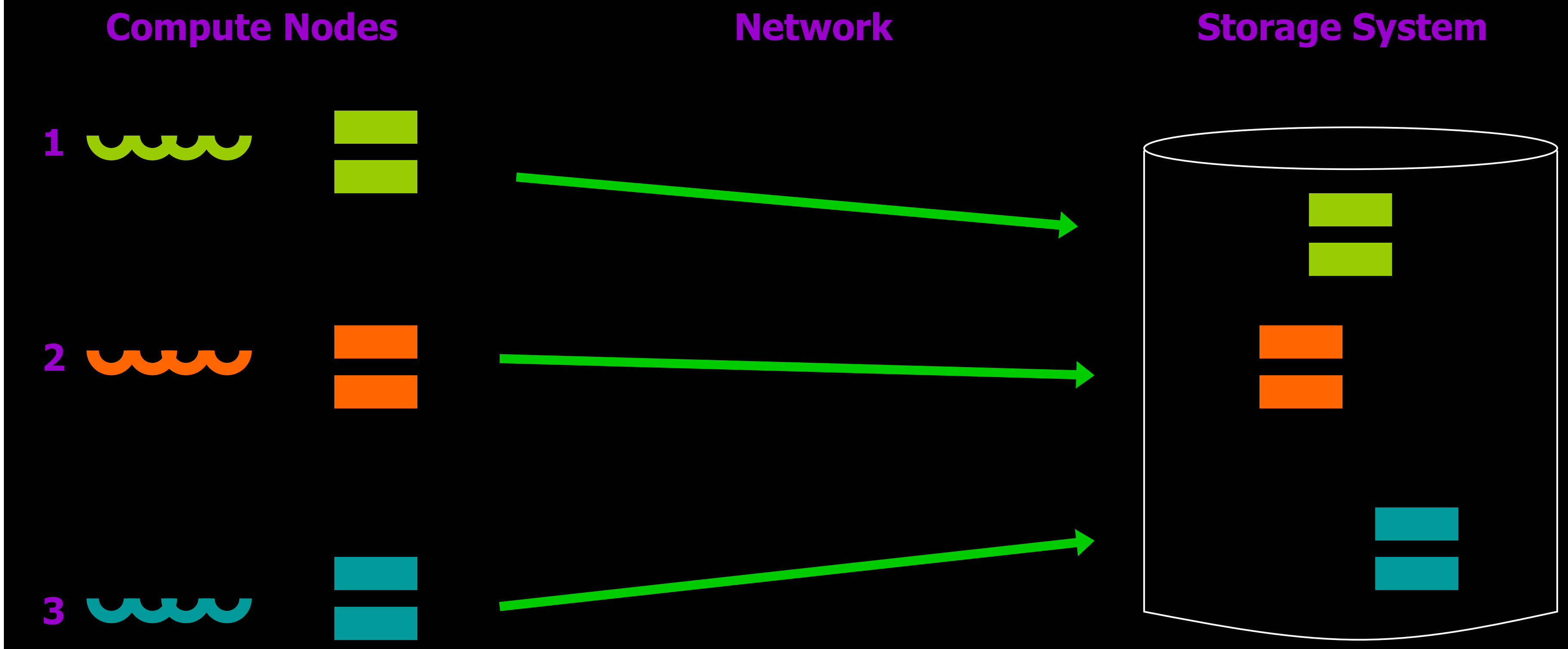
Sync-Barrier-Sync

3. SYNC: Perform file revalidation by ensuring written data is visible to all nodes.



Sync-Barrier-Sync

Nodes read new data



Agenda

Motivation

pNFS

HPC consistency requirements

Protocol consistency semantics

NFSv3 ADIO driver

pNFS ADIO driver

Protocol consistency semantics

	Revalidation	Data Flush
POSIX Last writer wins	Open Read	Fsync Close
MPI-IO (non-atomic)	MPI_File_Sync MPI_File_Open	MPI_File_Sync MPI_File_Close
pNFS Close-to-open	Open Fnctl lock “ ______ “	Close Fnctl ulock Fsync

Notes:

- POSIX does not require revalidation primitive
- NFS lacks primitive to leverage per-object change attribute

Agenda

Motivation

pNFS

HPC consistency requirements

Protocol consistency semantics

NFSv3 ADIO driver

pNFS ADIO driver

NFSv3 ADIO driver

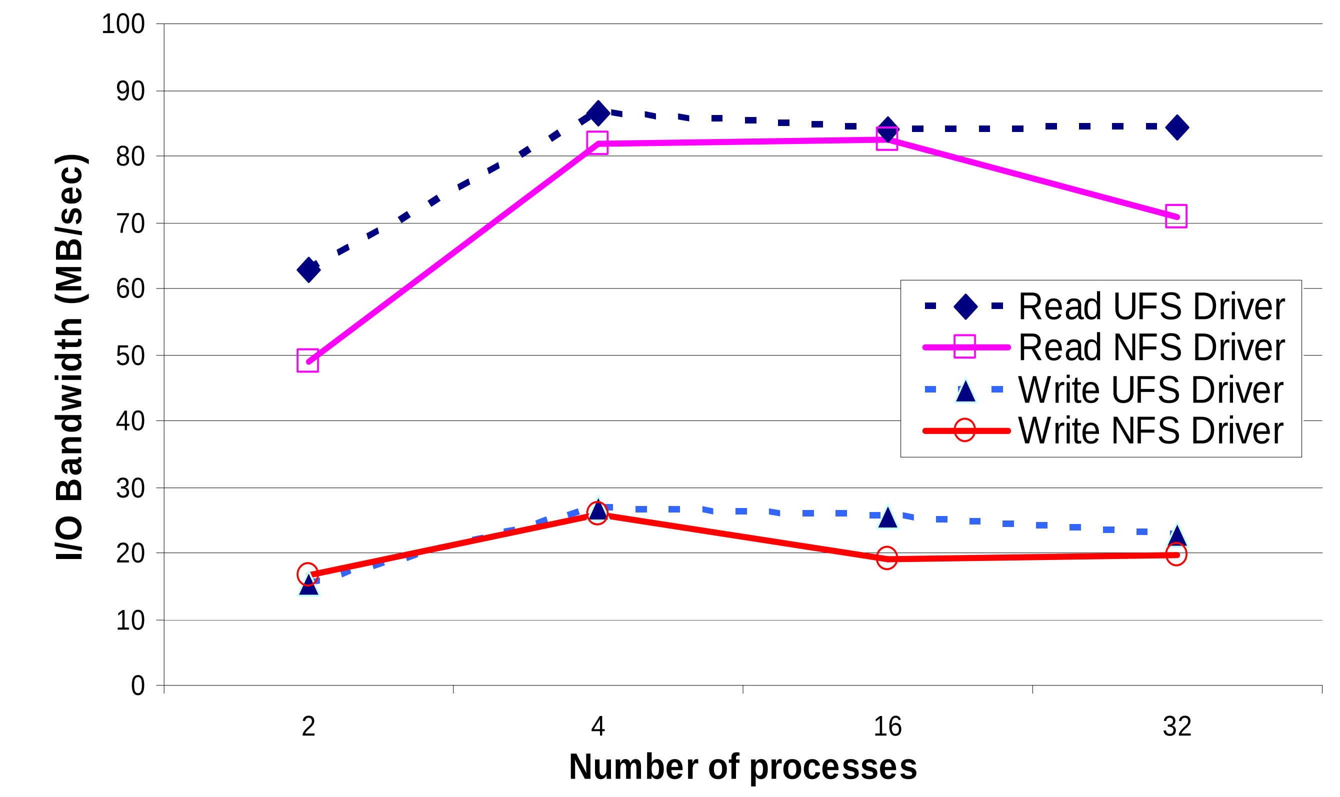
ROMIO/ADIO

- ROMIO is an I/O implementation of MPI-IO
- ADIO is a portable parallel I/O API that allows file systems to implement MPI-IO semantics

- NFSv3 ADIO driver
  - Forcing NFSv3 to comply with MPI-IO semantics hurts performance
  - Performs multiple close/open or lock/locku to revalidate file
    - Inconsistent NFSv3 implementations
    - Protocol and implementation problems with the NFSv3 lockd daemon.
    - Disallows attribute caching

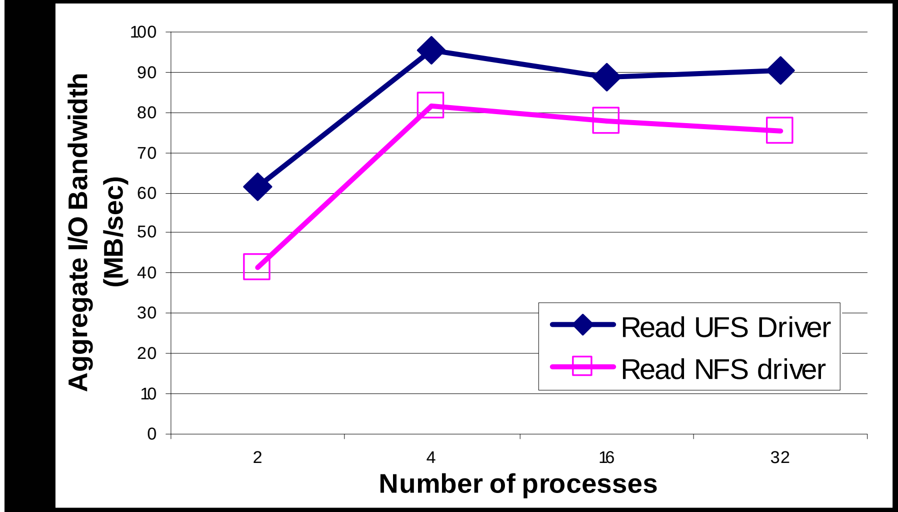
- UFS ADIO driver
  - POSIX compliant file systems

POP-IO NFSv4/Ext3 Read and Write

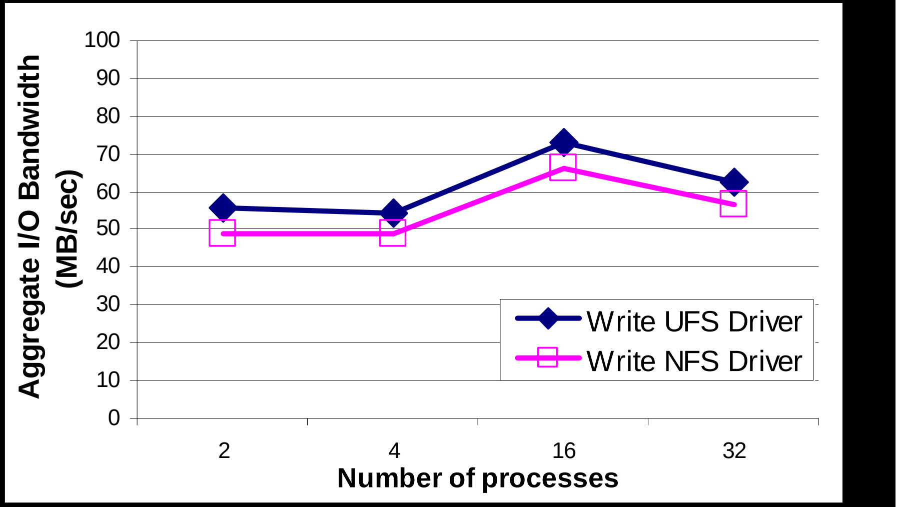


POP-IO pNFS/GPFS Read and Write

READ



WRITE



Agenda

Motivation

pNFS

HPC consistency requirements

Protocol consistency semantics

NFSv3 ADIO driver

pNFS ADIO driver

Looking forward: pNFS ADIO driver possibilities

- Some I/O workloads, e.g., checkpointing, require minimal data coherence
  - Try to optimize "all read" or "all write" workloads
- For applications that perform sync-barrier-sync:
  - HEC POSIX extensions for HPC
    - O_LAZY, lazyio_propagate(), lazyio_synchronize()
  - Direct I/O
    - No read or writeback cache
    - Possibly only on read path
  - Non-portable techniques:
    - Manually invalidate entire page cache
    - Fadvise
  - open/close and/or lock/ulock
    - Data must be written to disk
  - User-space client with customized interface
    - Support?
  - Others?

Summary

Good: MPI-IO and pNFS share similar relaxed semantics

Bad: HPC apps require on-demand file sync and revalidation (Lazy I/O)

Ugly: Interface to file system through POSIX interface

Lacks on-demand file revalidation

Need to investigate possible workarounds

Thank You!