

HPC I/O and File Systems Issues and Perspectives

Gary Grider, LANL

LA-UR-06-0473

01/2006

Why do we need so much HPC

□ Urgency and importance of Stockpile

Stewardship mission and schedule demand multi-physics predictive simulations

- supported on unprecedented terascale computing

□ Without nuclear testing we require validated simulations that have

- much greater physics fidelity,
- much higher resolution, and
- must be fully three dimensional.

What do our Applications do?



□ Nuclear Applications

- Weapons Performance and Output
- Safety
- Response in Abnormal and Normal Environments
- Design of Experiments and Experimental Facilities

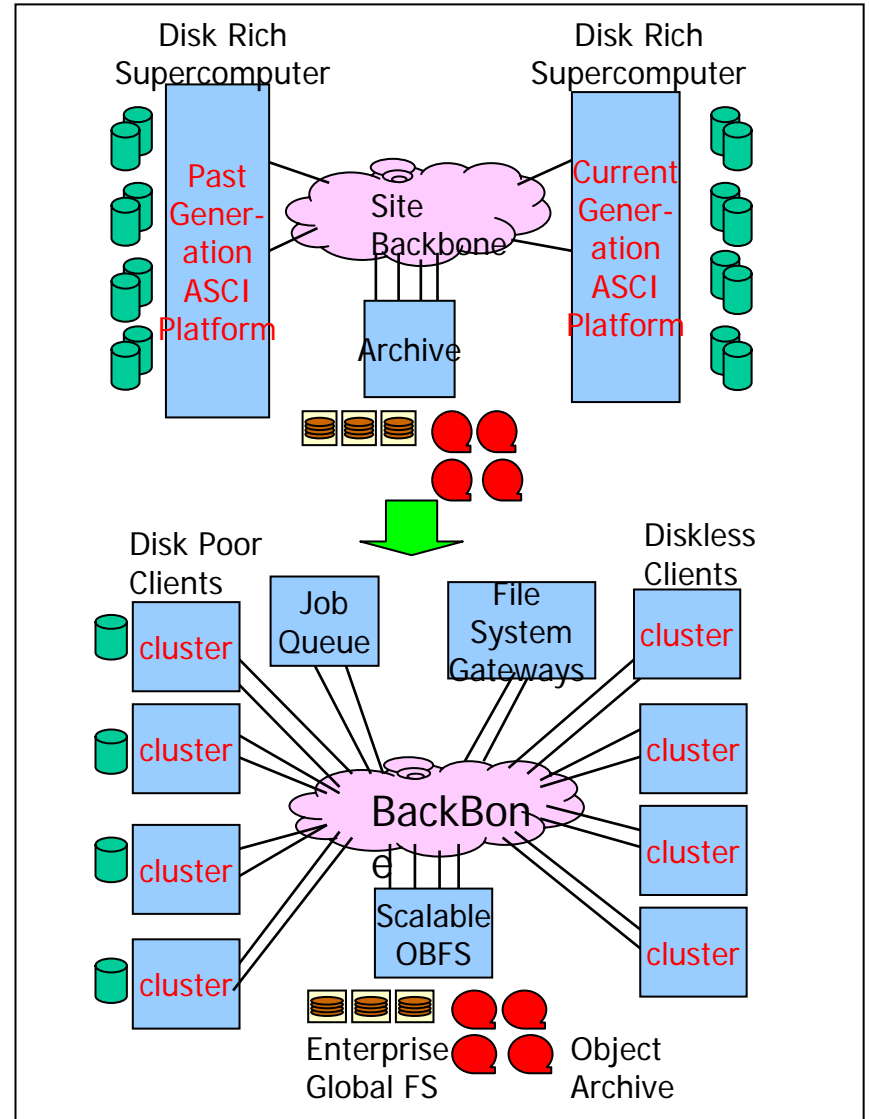
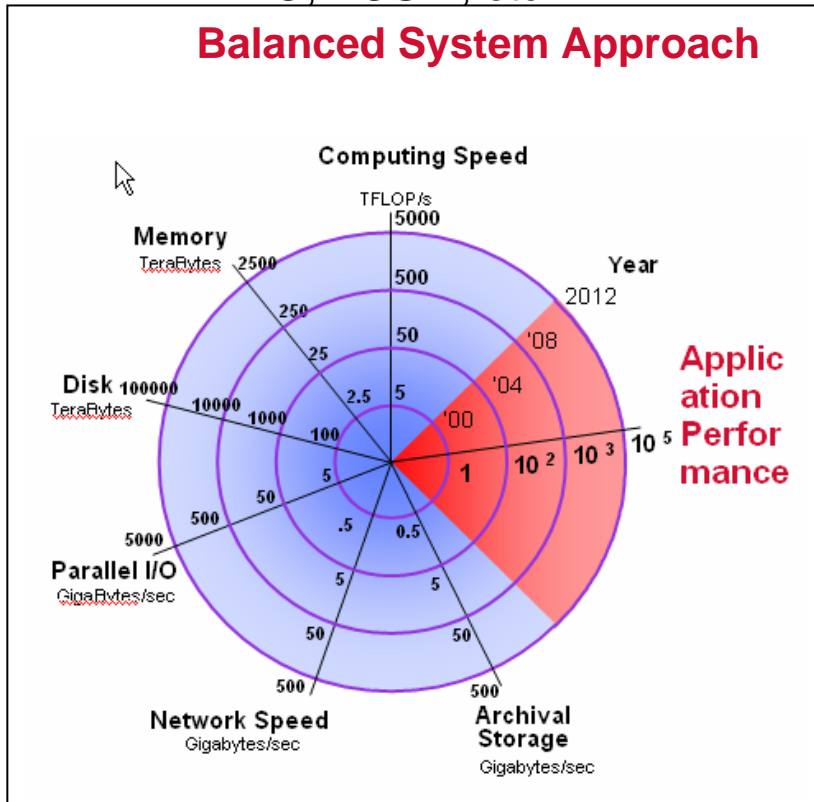
□ Non-Nuclear Applications

- Response in Abnormal and Normal Environments
 - Design and Manufacturing of Weapons Components
 - Design of Experiments and Experimental Facilities
-

What drives us?

- Provide reliable, easy-to-use, high-performance, scalable, and secure, I/O
- Via standard and other interfaces
 - MPI-IO, POSIX, etc.

Balanced System Approach



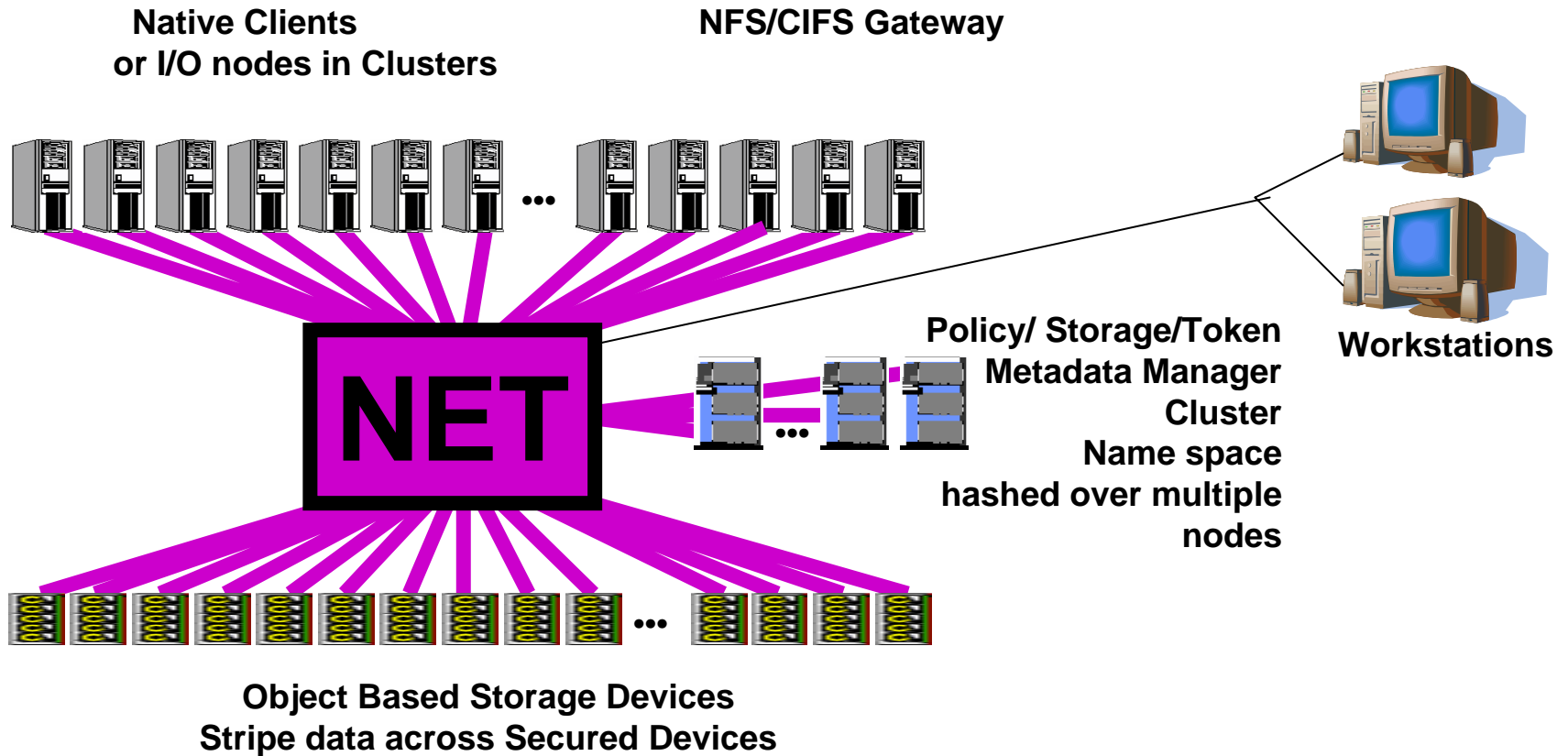
Market Drivers



- **The market seems to be heading us in three directions that cover the spectrum of machine paradigms**

- For capability runs (3D, 6 months to 2 years, nearly full machine)
 - Blue Gene/Red Storm like architectures with 10's to 100's of thousands of compute nodes
 - Lots of processors and nodes (100,000)
 - Thinner interconnect
 - Small memorys per processor
 - Accelerators where you might anticipate a teraflop sized compute node
 - Way fewer processors with lots of units per processor
 - Way fewer nodes (1000)
 - Extremely thick interconnect (IB 12X QDR)
 - Maybe very large memory and not terribly uniform per node
- For capacity runs (2D, few hours to a few days, 1/8th of full machine)
 - Standard clusters
 - Growing in number of nodes (1000-4000)
 - More cores
 - Commodity interconnects (IB 4X)

Parallel Object File Systems are Evolving



Requirements Summary

FS Requirements Summary



- ❑ From Tri-Lab File System Path Forward RFQ (which came from the Tri-labs file systems requirements document)

<ftp://ftp.lanl.gov/public/ggrider/ASCIFSRFP.DOC>

- *POSIX-like Interface, Works well with MPI-IO, Open Protocols, Open Source (parts or all), No Single Point Of Failure , Global Access*
 - *Global name space, ...*
 - *Scalable bandwidth, metadata, management, security*
...
 - *WAN Access, Global Identities, Wan Security, ...*
 - *Manage, tune, diagnose, statistics, RAS, build, document, snapshot, ...*
 - *Authentication, Authorization, Logging, ...*
-

FS Requirements Detail



- ❑ 3.1 POSIX-like Interface
- ❑ 3.2 No Single Point Of Failure
- ❑ 4.1 Global Access
 - 4.1.1 Global Scalable Name Space
 - 4.1.2 Client software
 - 4.1.3 Exportable interfaces and protocols
 - 4.1.4 Coexistence with other file systems
 - 4.1.5 Transparent global capabilities
 - 4.1.6 Integration in a SAN environment
- ❑ 4.2 Scalable Infrastructure for Clusters and the Enterprise
 - 4.2.1 Parallel I/O Bandwidth
 - 4.2.2 Support for very large file systems
 - 4.2.3 Scalable file creation & Metadata Operations
 - 4.2.4 Archive Driven Performance
 - 4.2.5 Adaptive Prefetching
- ❑ 4.3 Integrated Infrastructure for WAN Access
 - 4.3.1 WAN Access To Files
 - 4.3.2 Global Identities
 - 4.3.3 WAN Security Integration
- ❑ 4.4 Scalable Management & Operational Facilities
 - 4.4.1 Need to minimize human management effort
 - 4.4.2 Integration with other management tools
 - 4.4.2 Integration with other Management Tools
 - 4.4.3 Dynamic tuning & reconfiguration
 - 4.4.4 Diagnostic reporting
 - 4.4.5 Support for configuration management
 - 4.4.6 Problem determination GUI
 - 4.4.7 User statistics reporting
 - 4.4.8 Security management
 - 4.4.9 Improved Characterization and Retrieval of Files
 - 4.4.10 Full documentation
 - 4.4.11 Fault Tolerance, Reliability, Availability, Serviceability (RAS)
 - 4.4.12 Integration with Tertiary Storage
 - 4.4.13 Standard POSIX and MPI-IO
 - 4.4.14 Special API semantics for increased performance
 - 4.4.15 Time to build a file system
 - 4.4.16 Backup/Recovery
 - 4.4.17 Snapshot Capability
 - 4.4.18 Flow Control & Quality of I/O Service
 - 4.4.19 Benchmarks
- ❑ 4.5 Security
 - 4.5.1 Authentication
 - 4.5.2 Authorization
 - 4.5.3 Content-based Authorization
 - 4.5.4 Logging and auditing
 - 4.5.5 Encryption
 - 4.5.6 Deciding what can be trusted

Lots of things have to scale



File System Attributes

	1999	2002	2005	2008
Teraflops	3.9	30	100	400
Memory size (TB)	2.6	13-20	32-67	44-167
File system size (TB)	75	200 - 600	500 -2,000	20,000
Number of Client Tasks	8192	16384	32768	65536
Number of Users	1,000	4,000	6,000	10,00
Number of Directories	$5.0 \cdot 10^6$	$1.5 \cdot 10^7$	$1.8 \cdot 10^7$	$1.8 \cdot 10^7$
Metadata Rates Data Rate	500/sec 1 mds 3 GB/sec	2000/sec 1 mds 30 GB/sec	20,000/sec n mds 100 GB/sec	50,000/sec n mds 400 GB/sec
Number of Files	$1.0 \cdot 10^9$	$4.0 \cdot 10^9$	$1.0 \cdot 10^{10}$	$1.0 \cdot 10^{10}$

Other Requirements

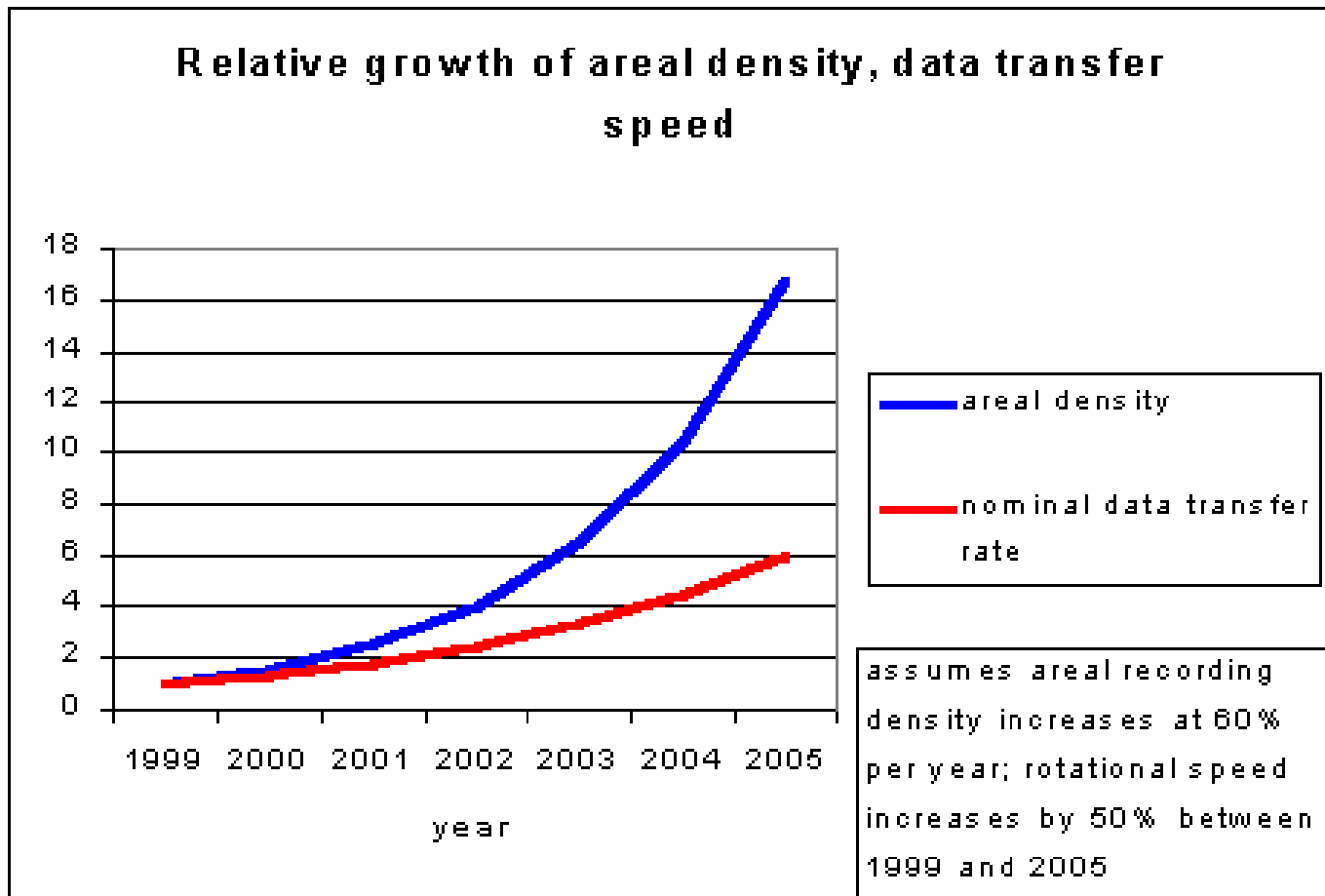
- Based on Standards**
 - Security**
 - Content based security, born on marks, hooks for end to end encryption, extensible attributes, etc.
 - Real transactional security on the SAN, not simple zoning and other poor attempts (ANSI T10)
 - Global, Heterogeneous, Protocol Agnostic, open source, open protocols**
 - POSIX behavior with switches to defeat portions**
 - Lazy attributes, byte range locks, etc.
 - WAN behavior like AFS/DFS but better**
 - Including ACL's, GSS, multi domain, directory delegation, etc.
 - Scalable management (sorry, scalability keeps coming up)**
 - A product, supported by a market larger than the Tri-Labs**
-

Trends and Emerging Issues

Emerging BW Issues

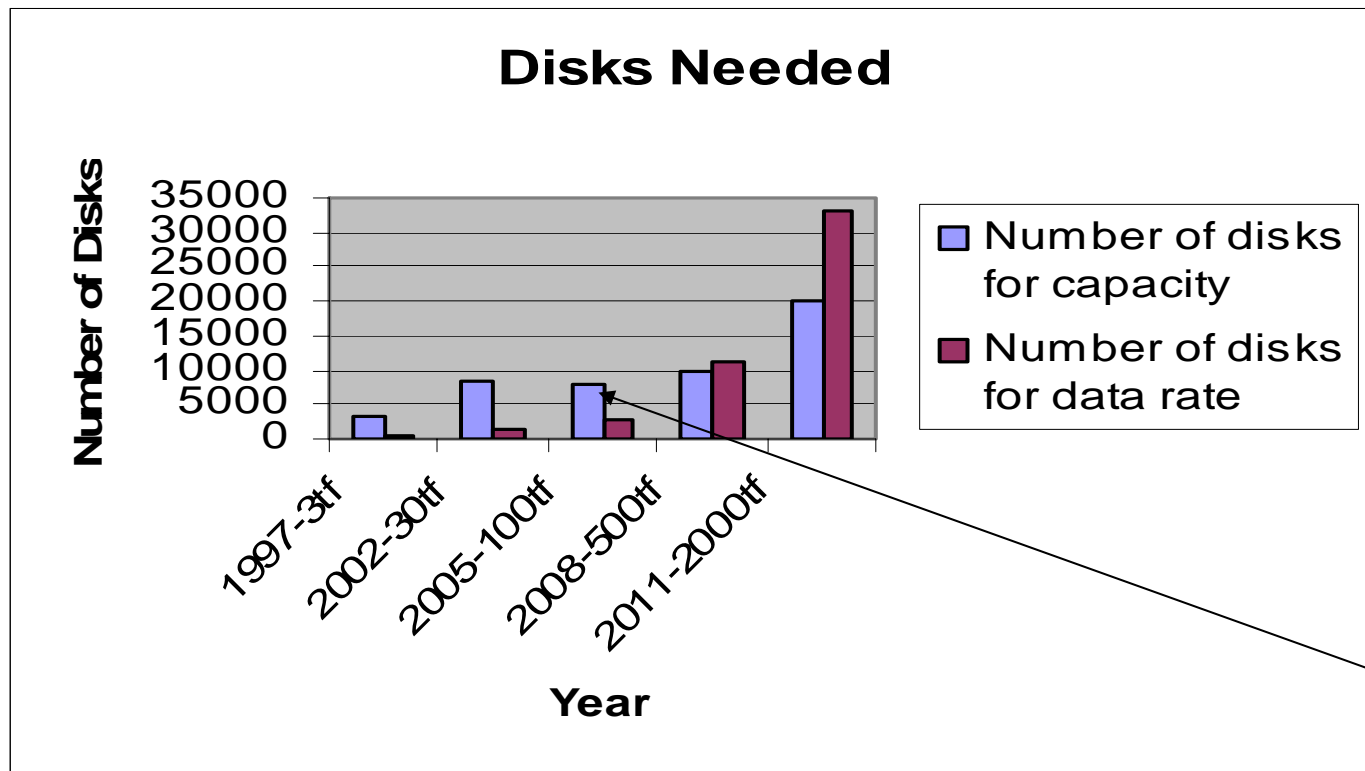


- ❑ Disks are getting much denser but not appreciably faster (bandwidth read/write)



Capacity vs BW Trend

- ❑ We will be buying disks for BW rather than for Capacity
- ❑ Write size for single disk sweet spot keeps rising
- ❑ Files will be striped over a larger percentage of the disks on the floor on average to get desired data rate
- ❑ We have to buy WAY more disks to get the BW!
- ❑ This has implications on reliability (WAY MORE DISKS)!

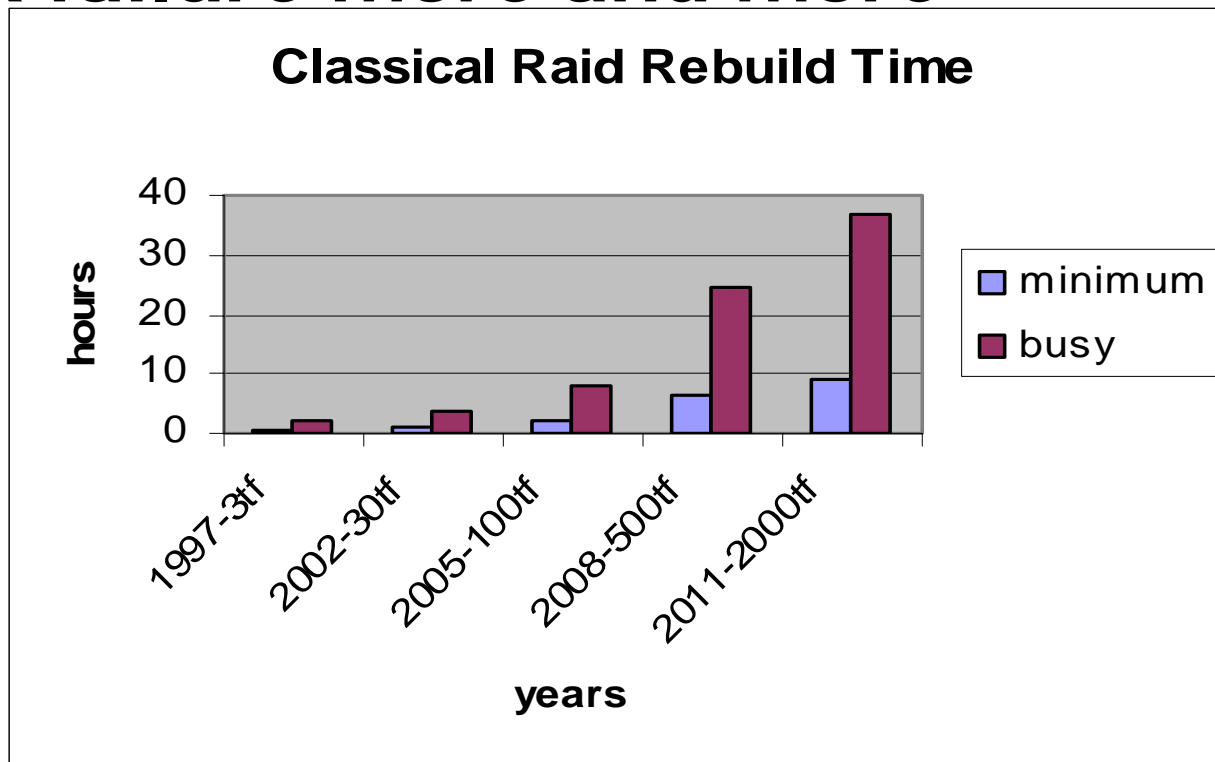


SATA

Classical RAID Rebuild Times

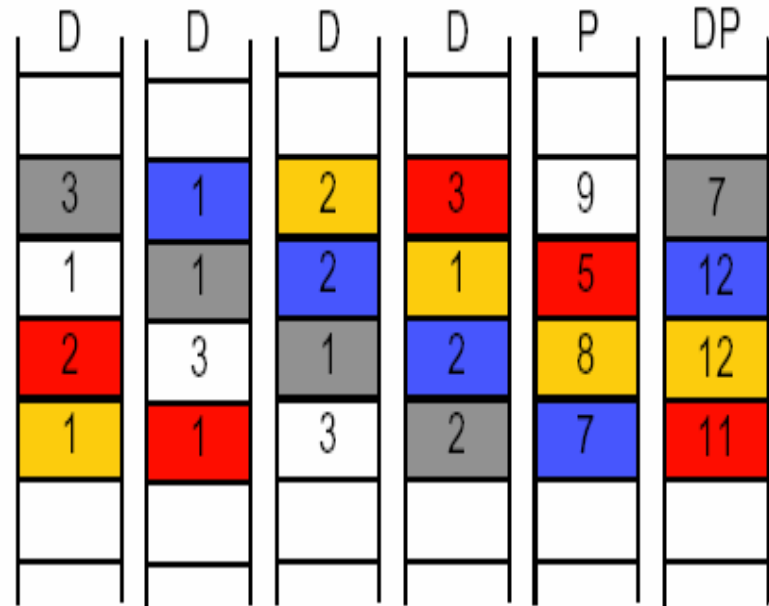
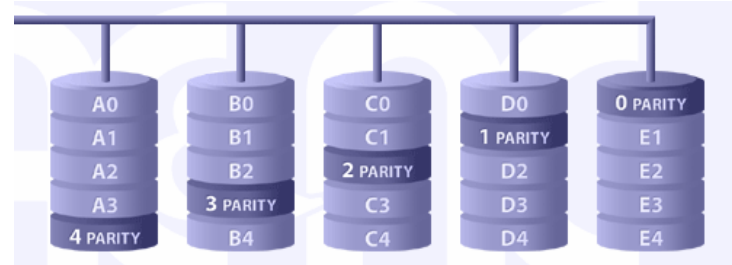


- Due to Capacity vs. BW, rebuild times get worse and worse, from minutes, to hours, to days – raising chances of 2-3 disk failure more and more



Will Plus N RAID technologies save us?

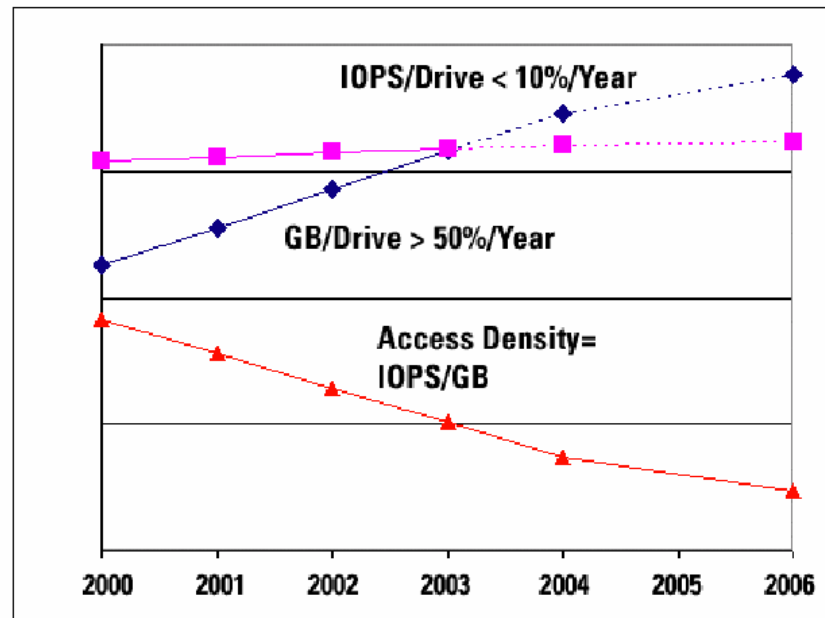
- ❑ Simple +1 XOR parity is calculated straight across the disk blocks
- ❑ Diagonal parity is calculated on diagonals, there are other methods based on polynomials
- ❑ There are other techniques for +N RAID
- ❑ The problem with +1 is that disk blocks are getting bigger so the amount of data required for an efficient write is getting larger
- ❑ +N technology makes this problem worse as it requires even more data for an efficient write
- ❑ Add the long rebuilt times issue, only partially mitigated by +N!
- ❑ **IS CONVENTIONAL +1 and +N RAID OUR ENEMY?**



Scalable Metadata

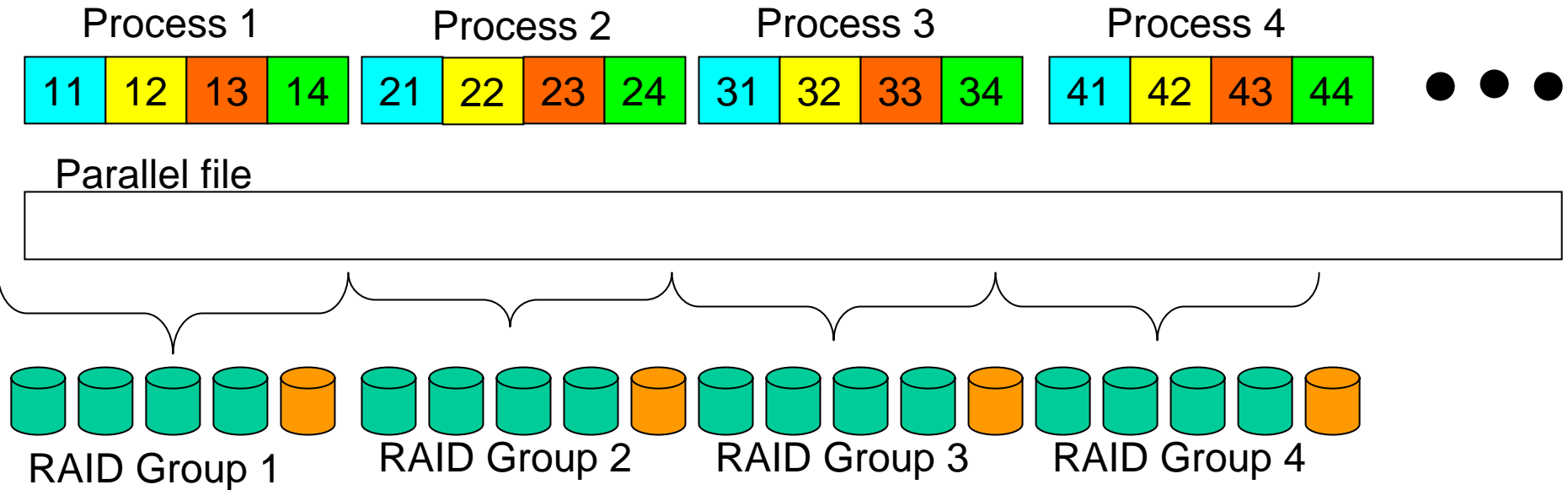


- ❑ Disks not getting more agile, Metadata services must scale
- ❑ Due to growth in global use from many clusters and due to usage patterns, N to N, N to 1 small ops, etc. metadata scaling issues are upon us.



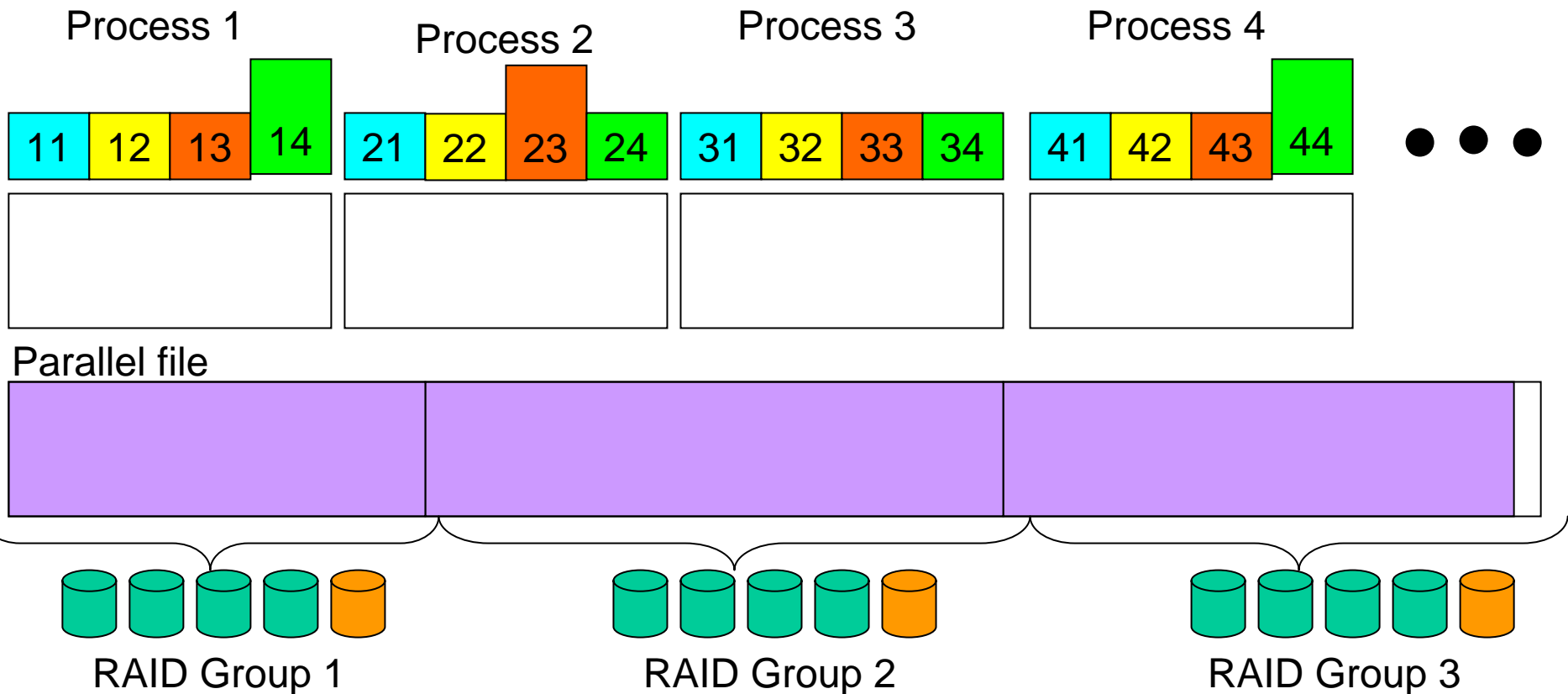
What kind of IO patterns do apps really present?

Example of well aligned I/O



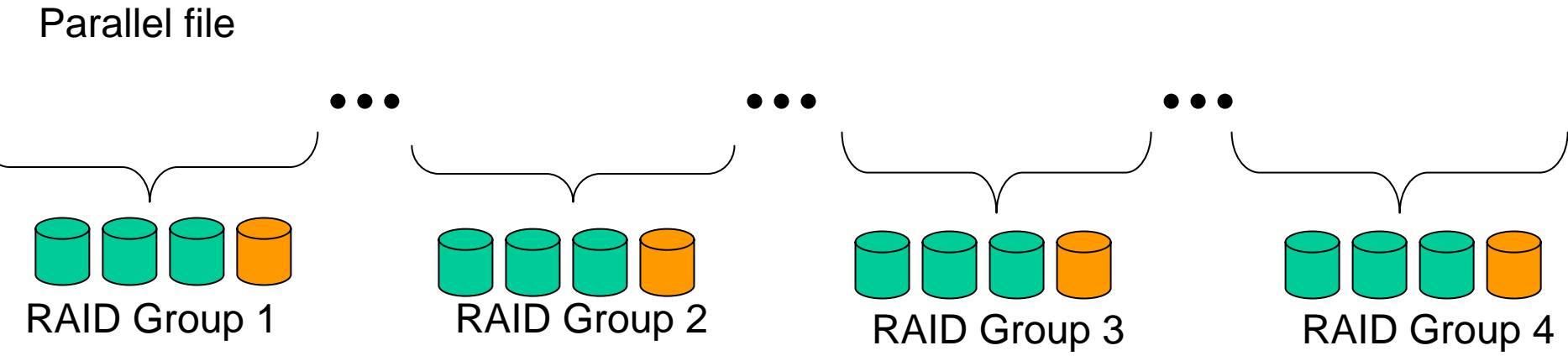
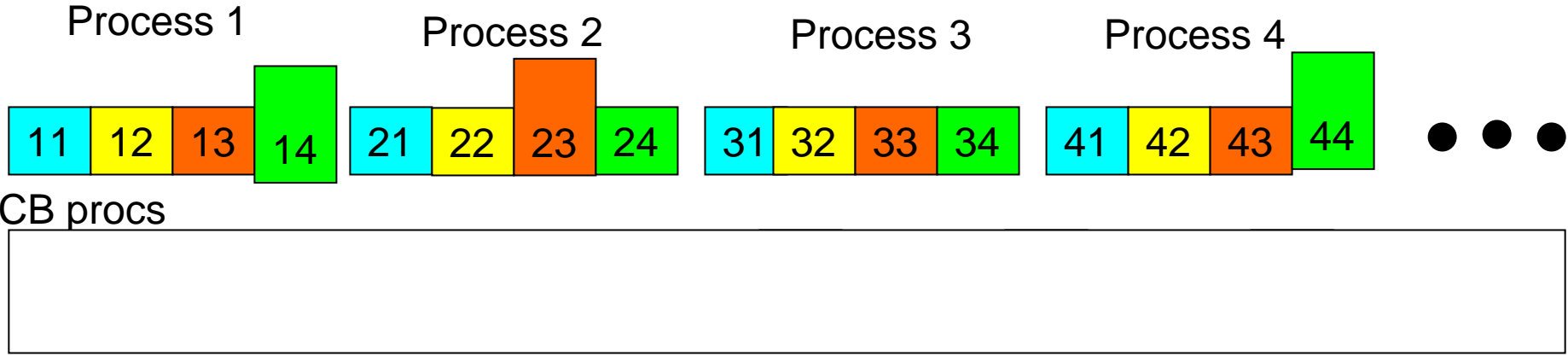
Oh, if applications really did I/O like this!

Real applications do small, unbalanced, and unaligned I/O

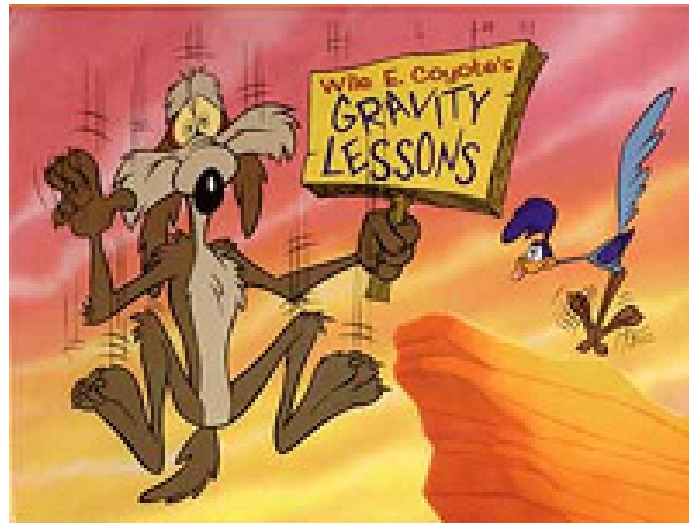


Notice every write is possibly a read/update/write since each write is a partial parity update. Notice that processes are serializing on their writes as well.

Middleware can help but more work is needed



One of the really hard problems!



Courtesy of Warner Bros. Entertainment Inc.

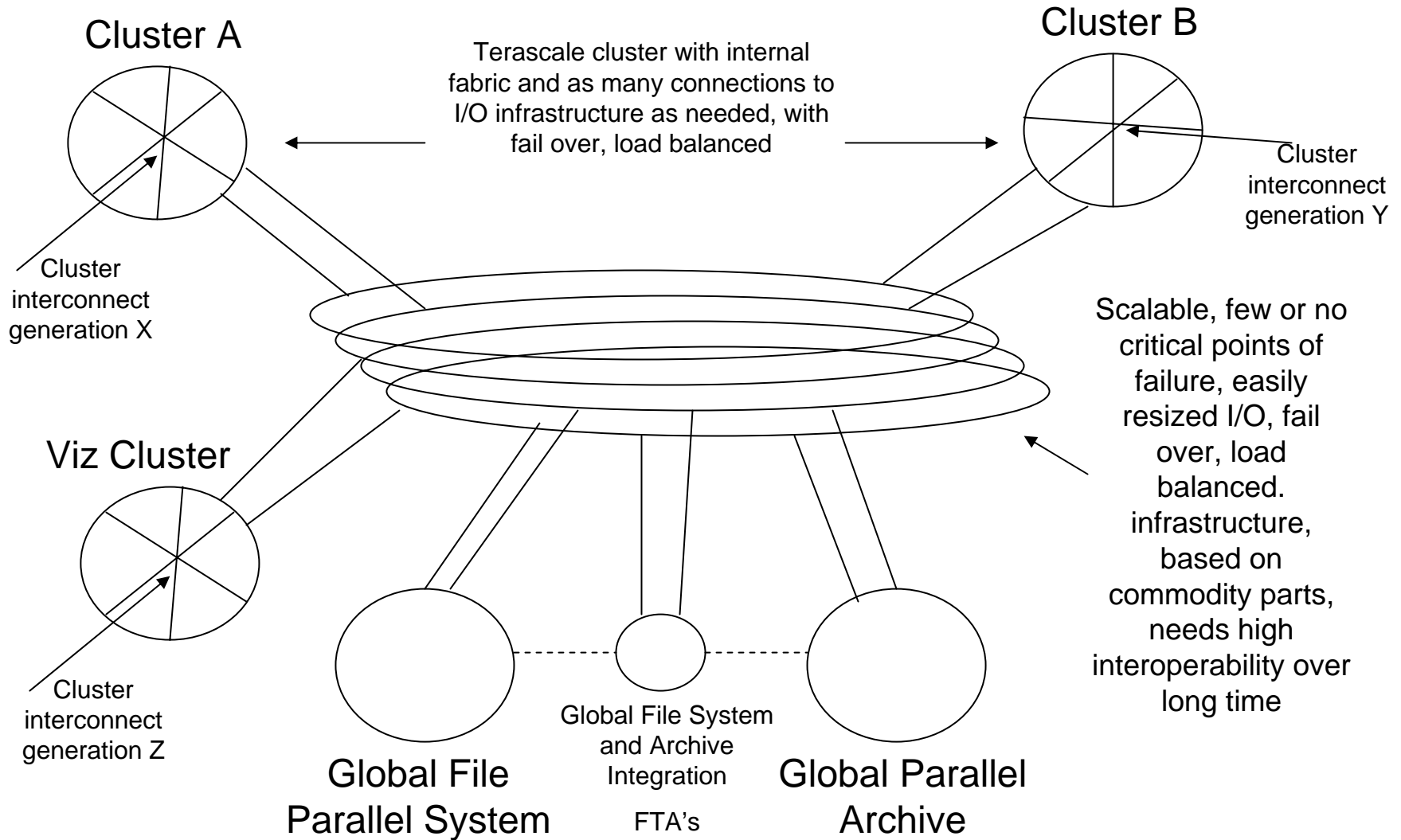
A Dilemma



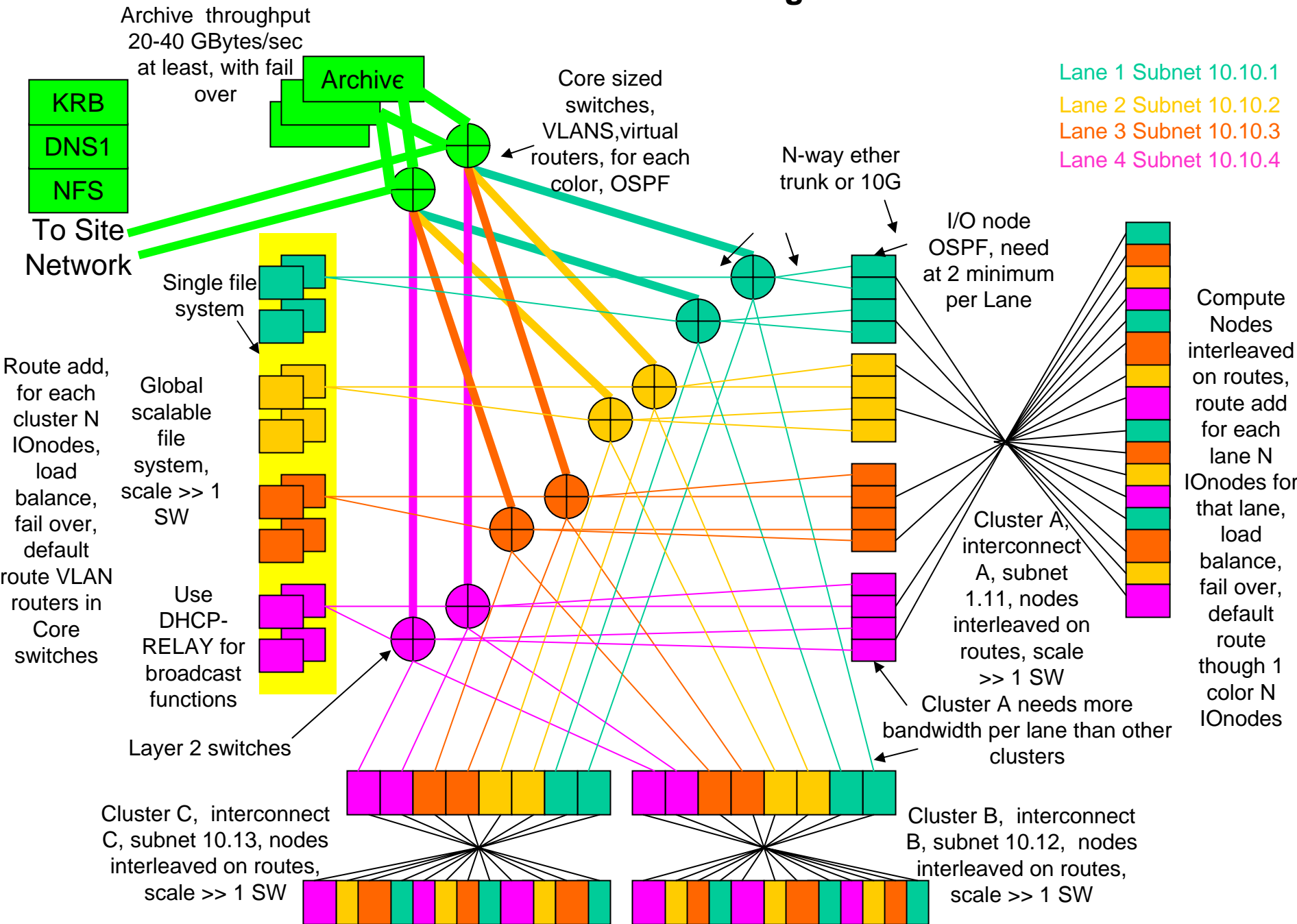
- It takes more and more disks to get the BW meaning we need more protection
 - Normal RAID approaches have increasing rebuild times
 - More protection under normal RAID approaches requires larger and larger writes for Efficiency
 - YET
 - APPS do not want to write larger and larger individual well aligned writes
-

Another hard problem: How do we get many terascale clusters connected to a global common parallel file system?

Where do we want to go, the bigger picture?



Current thinking



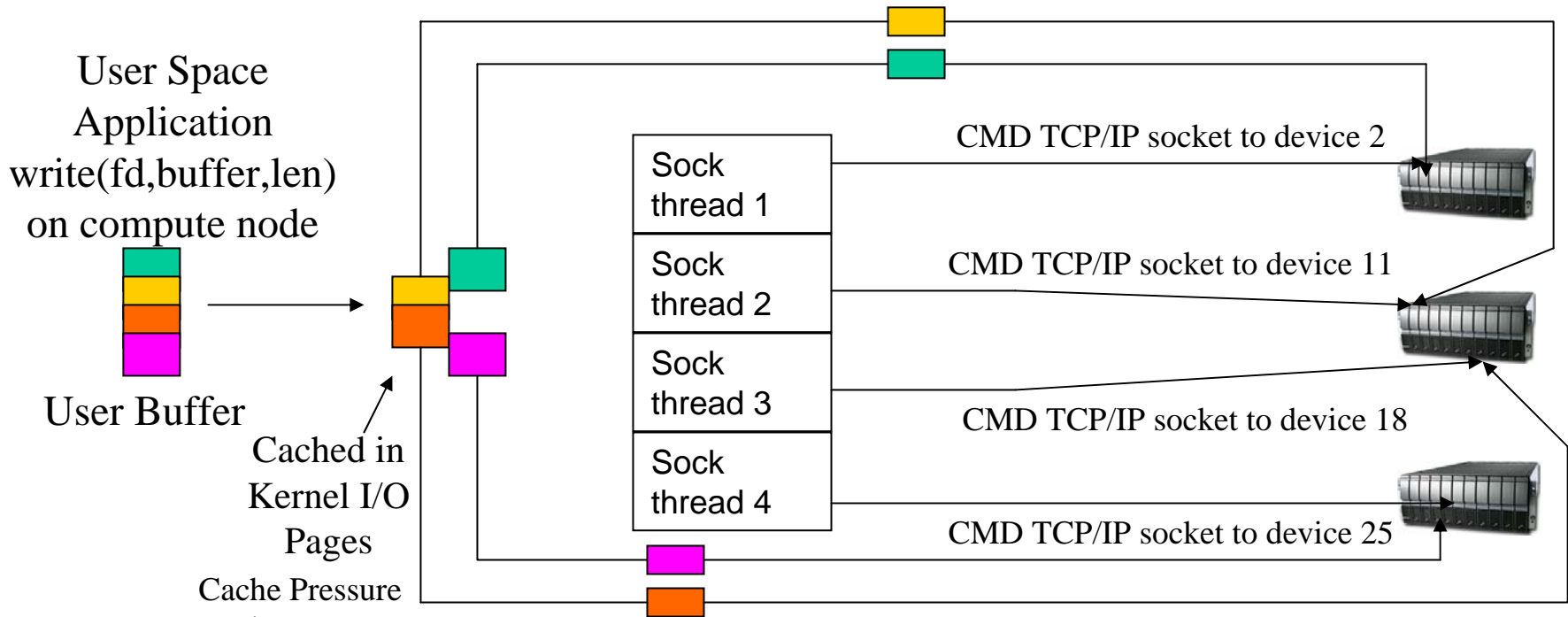
Some interesting initiatives!

POSIX IO Additions

- ❑ **Most of the current issues with POSIX IO semantics lie in the lack of support for distributed/parallel processes**
 - Concepts that involved implied ordering need to have alternative verbs that do not imply ordering
 - Vectored I/O read/write calls that don't imply ordering
 - Extending the end of file issues
 - Group opens
 - Etc.
 - Concepts that involve serialization for strict POSIX metadata query/update need to have lazy POSIX alternatives
 - Last update date (mtime, atime, ctime), Size of file
 - Active storage
 - ❑ **Status:**
 - Pursuing Labs joining "The Open Group" which holds the current "One Unix" charter which merges IEEE, ANSI, and POSIX standards and evolves the POSIX standard for UNIX
 - Next Step is to write up proposed new changes and begin discussion process within the POSIX UNIX IO API working group in the Open Group forum.
-

- Authored joint R&D needs document for next five years of needed I/O and file systems R&D work (DOE NNSA/Office of Science, DOD NSA)**
 - HEC/IWG I/O and File Systems Workshop and mechanism for coordinating Government funded R&D in this area**
 - First outcome, \$12M NSF call for R&D**
-

Server side pull via ISER



For large transfers the server (storage) remotely maps the clients memory via RDMA and moves the data for the client, completely offloaded and scheduled by the server (storage)

Small transfers

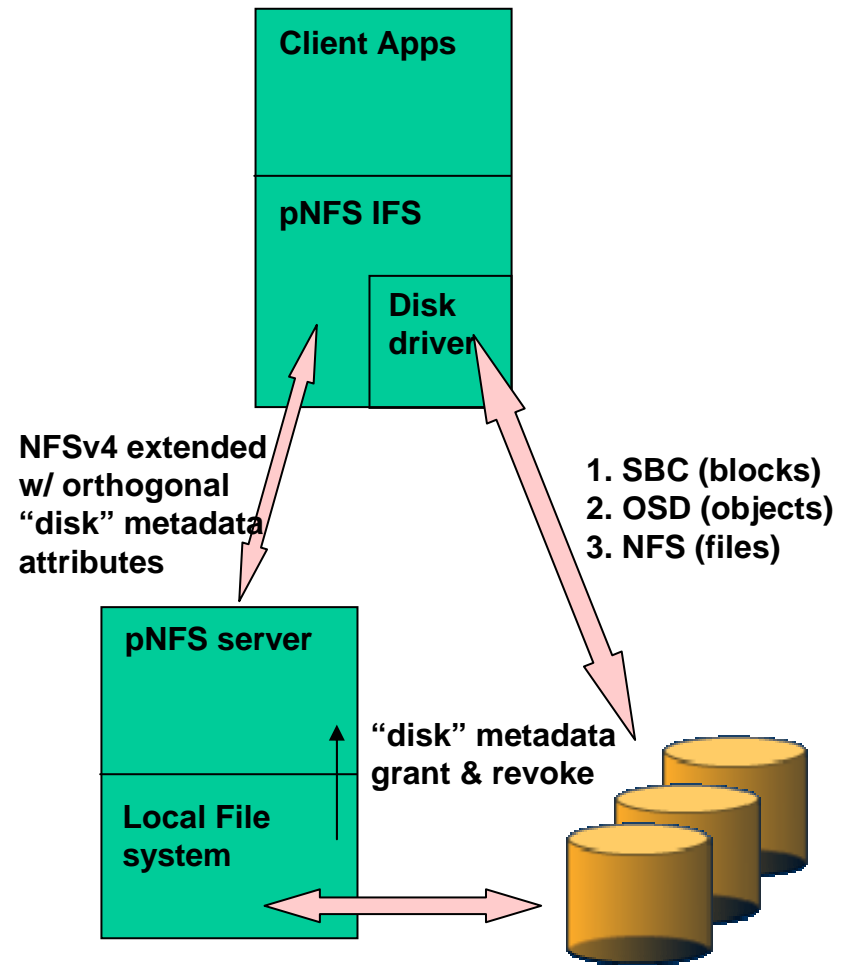
IETF IP info	IETF TCP info	ANSI T10 OSD iSCSI extension security capability Object ID variable length	IETF iSCSI CMD write (offset/length)	SCSI DATA Payload	Footers
--------------	---------------	--	--------------------------------------	-------------------	---------

Large transfer

IETF IP info	IETF TCP info	ANSI T10 OSD iSCSI extension security capability Object ID variable length	IETF iSCSI CMD write (offset/length)	Client remote mem map	Footers
--------------	---------------	--	--------------------------------------	-----------------------	---------

pNFS Multiple Data Server Protocols

- ❑ Inclusiveness favors success
- ❑ Three (or more) flavors of out-of-band metadata attributes:
 - **BLOCKS:** SBC/FCP/FC or SBC/iSCSI... for files built on blocks
 - **OBJECTS:** OSD/iSCSI/TCP/IP/GE for files built on objects
 - **FILES:** NFS/ONCRPC/TCP/IP/GE for files built on subfiles
- ❑ **Inode-level encapsulation** in server and client code



Object Archives

