HPC I/O and File Systems, is everyone out to get us?

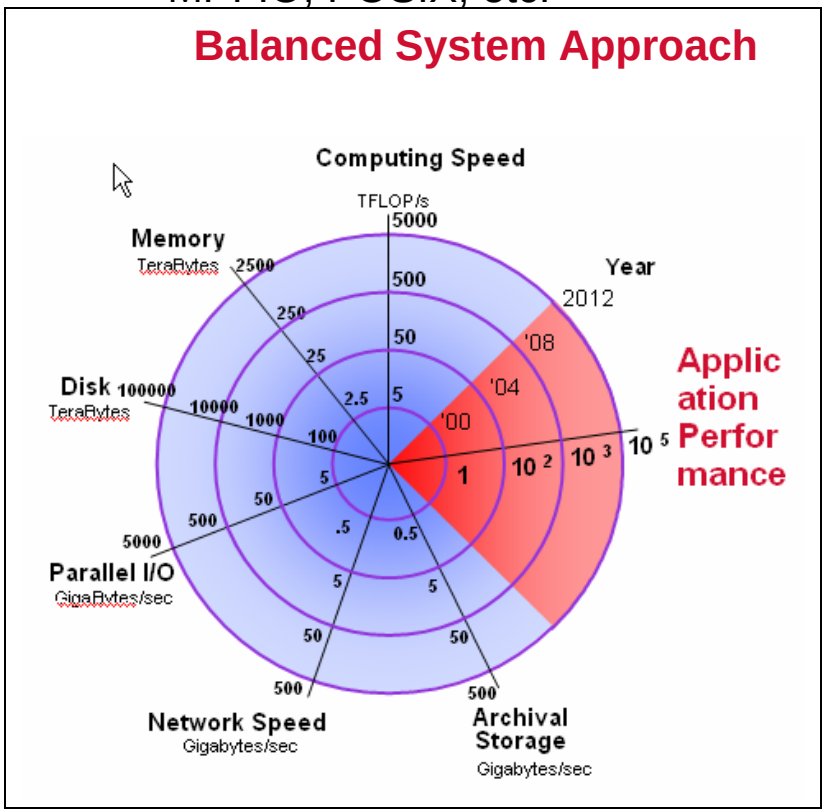
Gary Grider, LANL

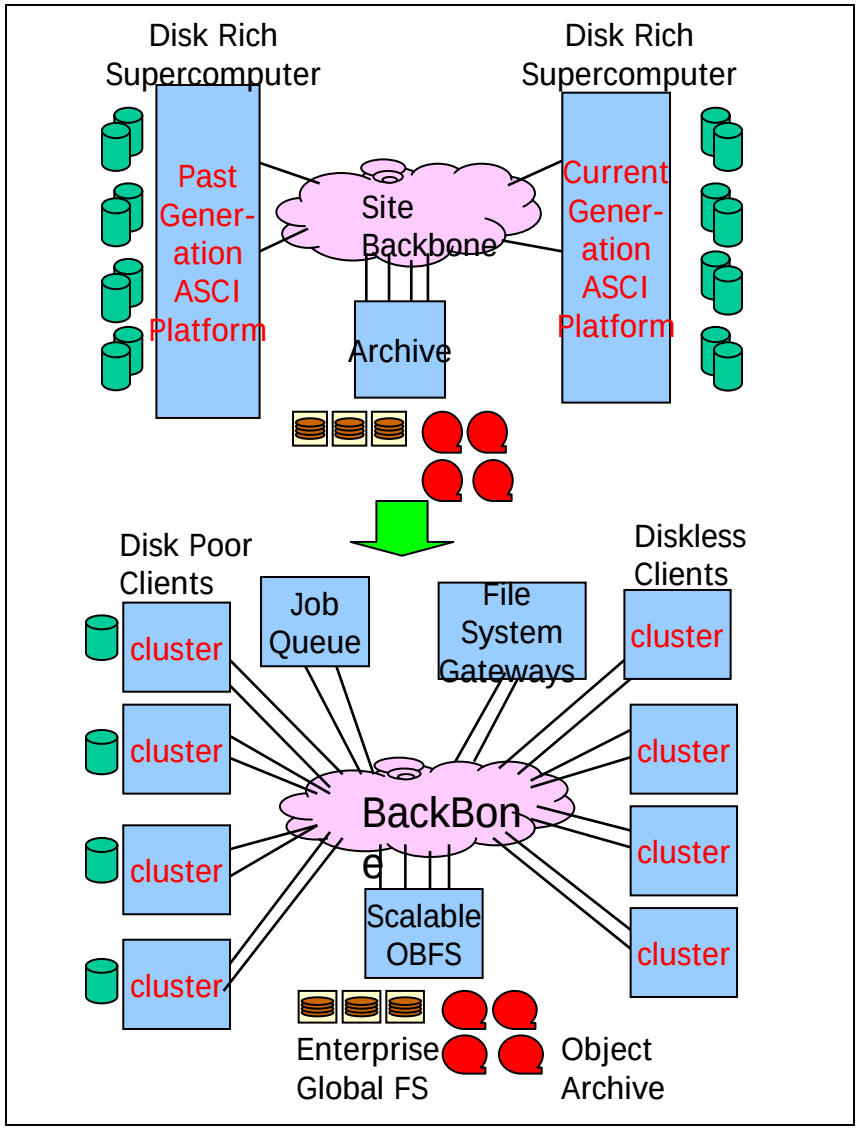
LA-UR-06-1122

02/2006

What drives us?

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





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*10^6	1.5*10^7	1.8*10^7	1.8*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*10^9	4.0*10^9	1.0*10^10	1.0*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**

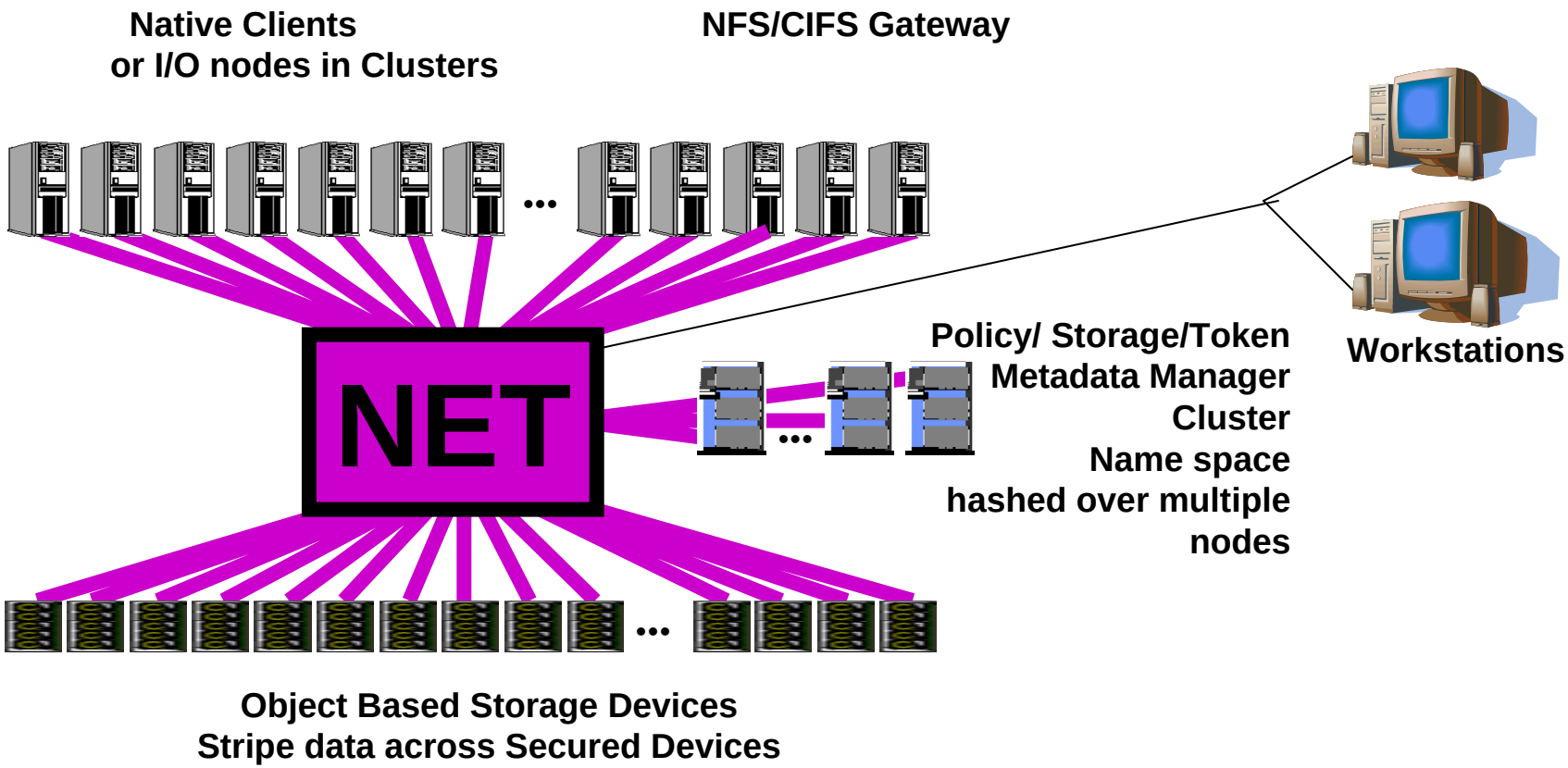
Seems easy enough, …

well maybe not!

First: A Tutorial

Why do I need to sit through a high level tutorial on File Systems?
To understand the problems we face.

Parallel Object File System

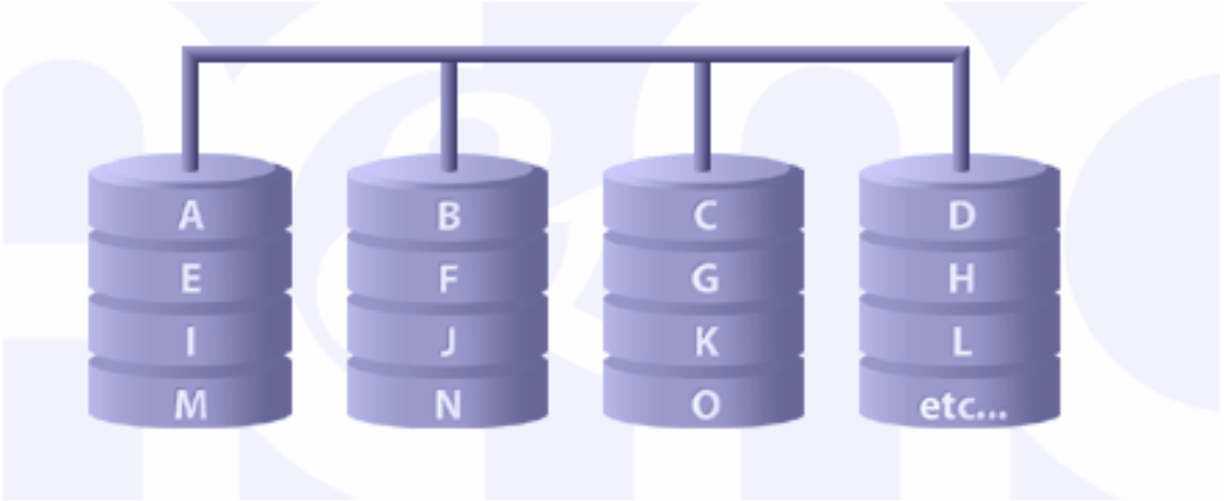


RAID

- Protection from data loss due to failing disks
- Periodic scrubbing of disks to detect failure quickly
- Disk problem/retry counting to detect failure quickly
- On the fly rebuild during normal traffic, hot sparing, notifications, etc.

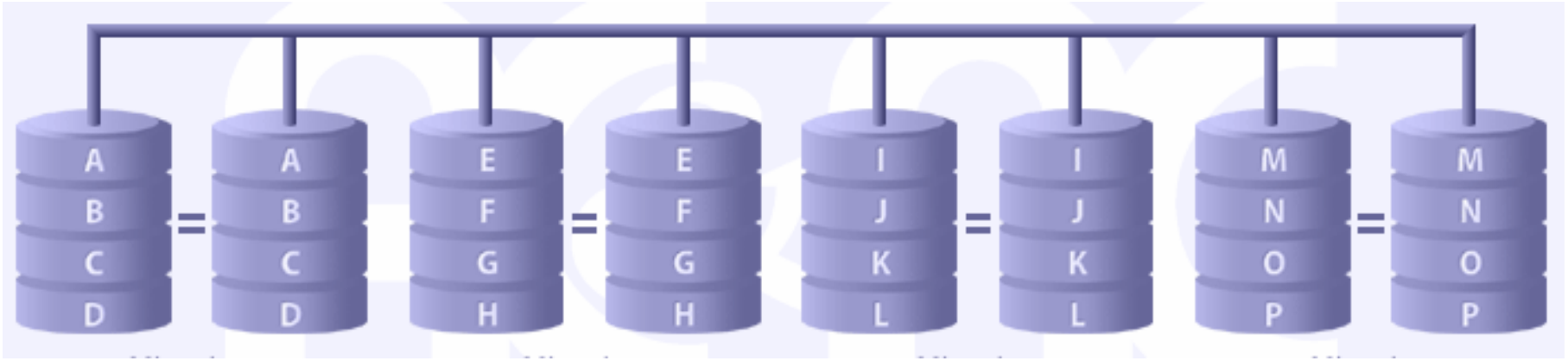
RAID 0

- Striped disk array without fault tolerance, i.e. no parity stripe.
- Data is broken down into blocks and each block is written to a separate storage blade.



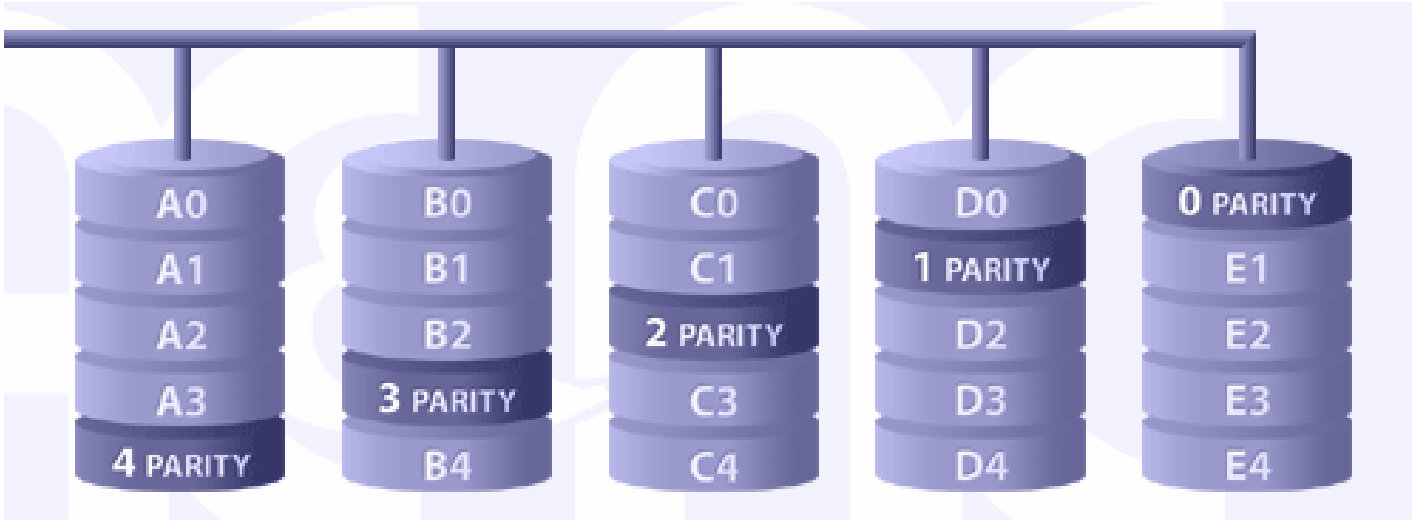
RAID 1

- Mirroring or writing the same data to two storage devices at once.



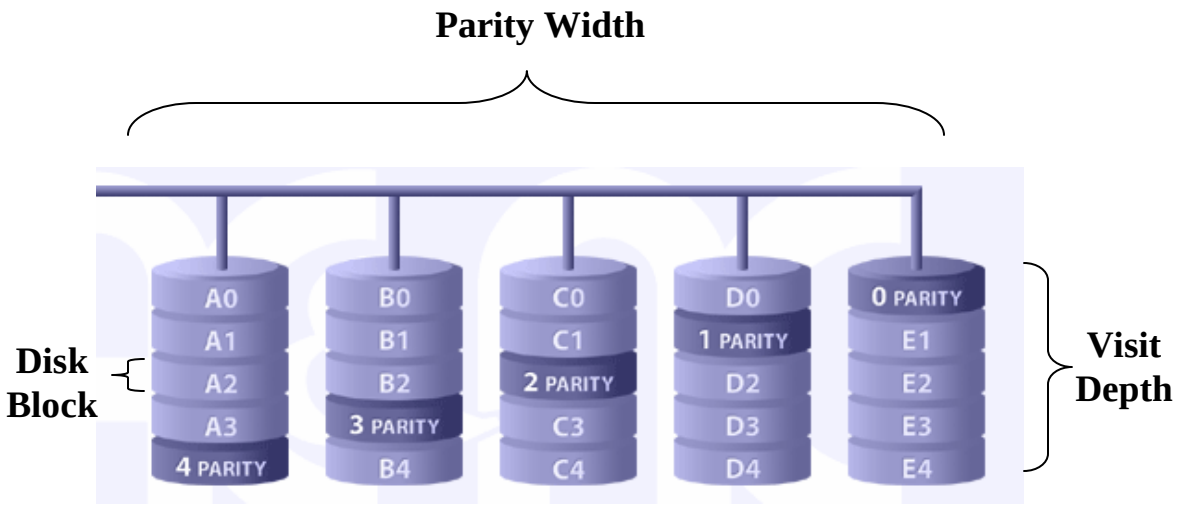
Plus 1 RAID 5

- Independent data disks with distributed parity blocks. Data is striped across a number of storage devices and a parity stripe is written for fault tolerance. Parity load is shared.



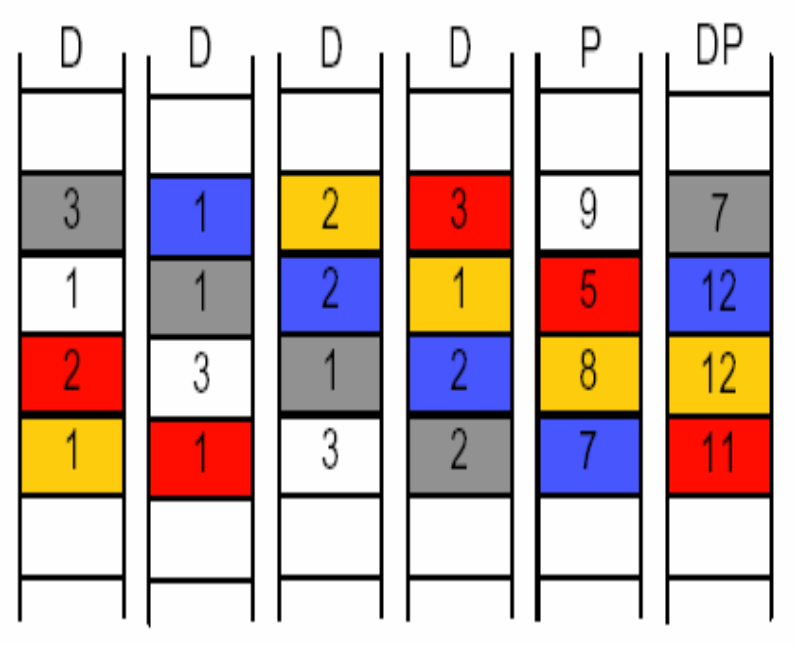
Plus 1 Layout – How to be efficient

- Disk Block (sweet spot for drive technology, 64k-256k, gets bigger with denser drives)
- Parity width N+P (typically 8+1 to 9+1 range
- Visit Depth (varies but for efficient pipe filling, think 100ish or more)
- For Efficient pre-calculated parity write operations for common RAID N+1 (N*block), think 1-2 MBytes and getting bigger
- To keep the pipe full, think 10's to 100 Mbyte sized operations to a Group of disks with parity



Plus 2 RAID – How to be efficient

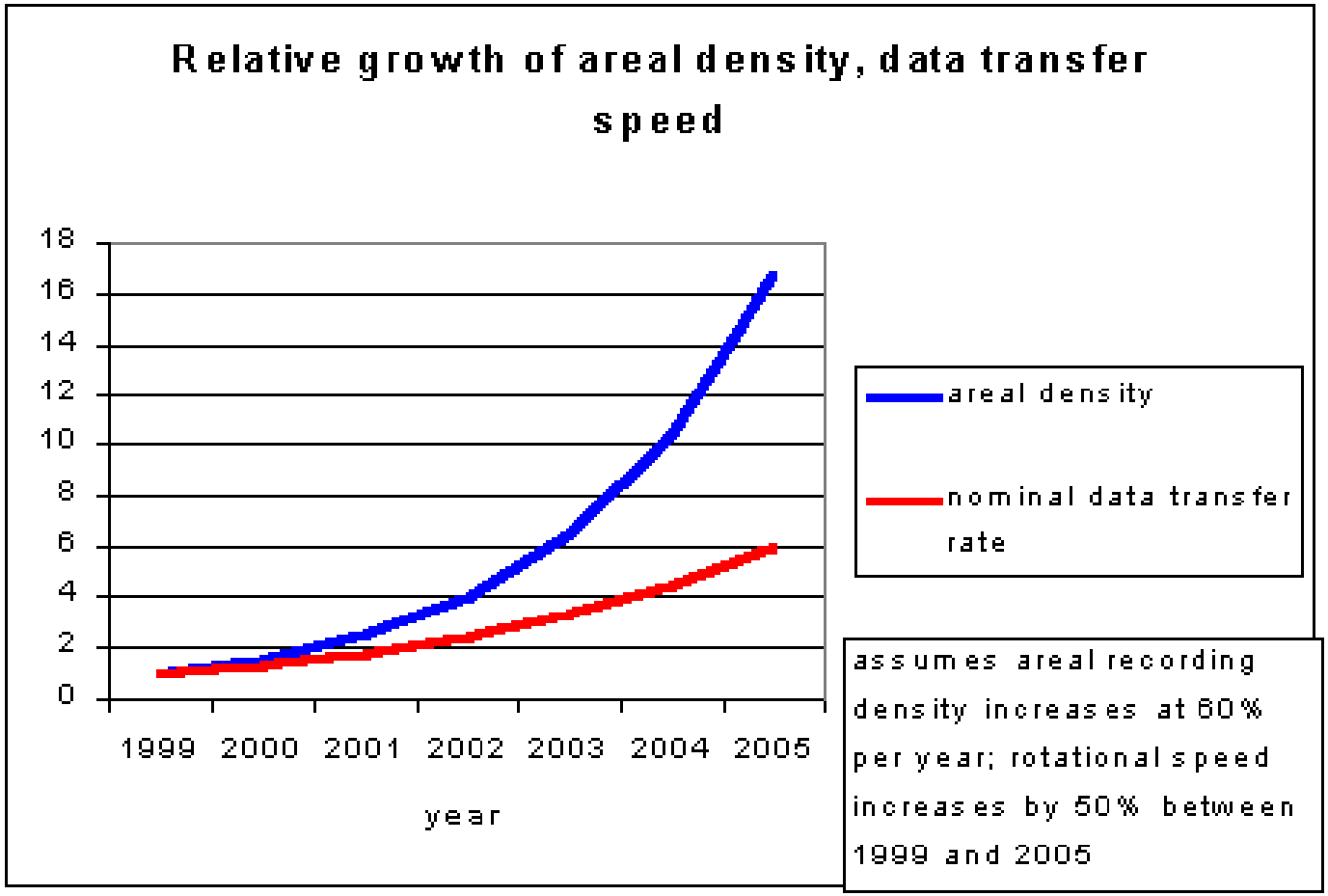
- Normal XOR parity is calculated straight across the disk blocks
- Diagonal parity is calculated on diagonals, there are other methods based on polynomials
- For Efficient pre-calculated parity write operations for common RAID N+1 (N^2 * block), think 8-16 MBytes and getting bigger
- To keep the pipe full, think 100 Mbytes or bigger to a RAID Group
- You need to have way more data around to do efficient parity calculation



Some interesting trends

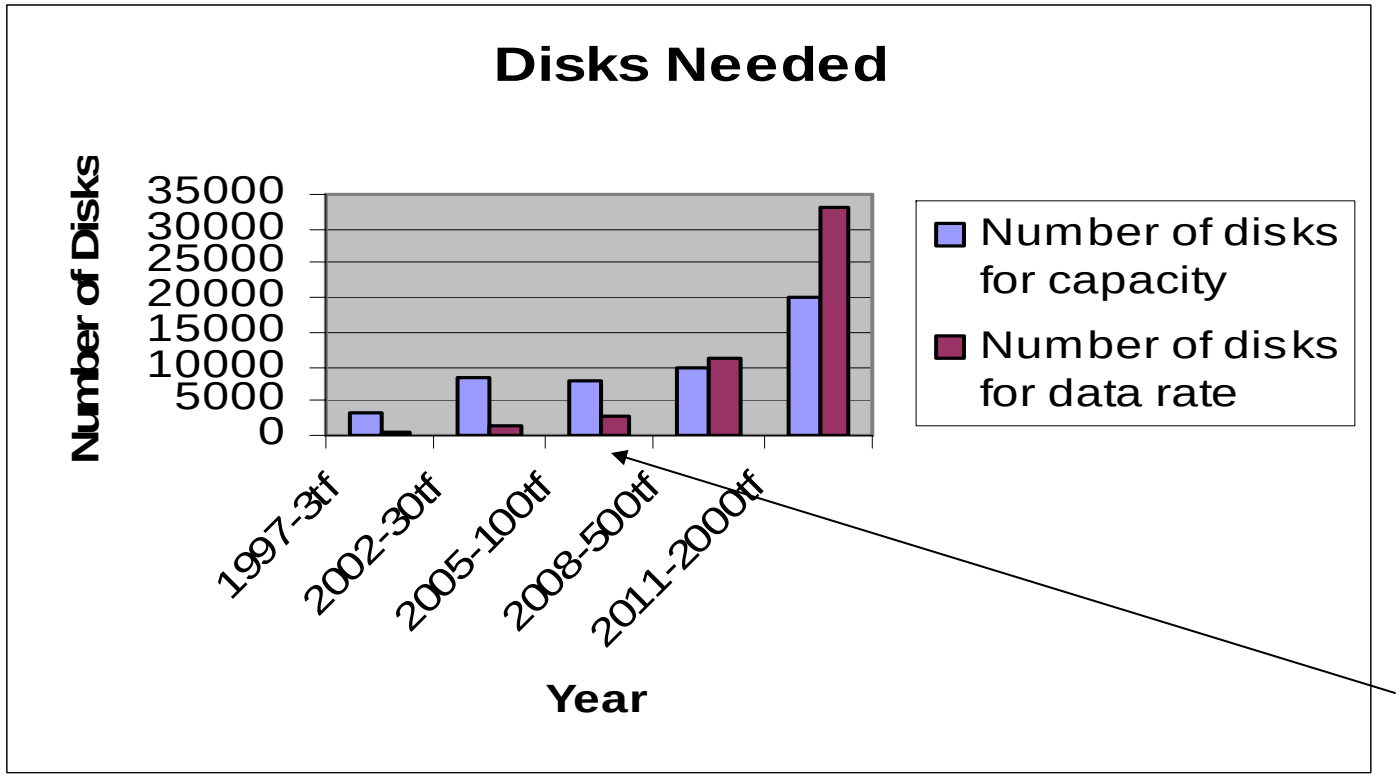
Emerging Issues with RAID

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



RAID Oriented Implications of Capacity vs BW Trend

- We will be buying more disks for BW than for Capacity
- Write size for single disk sweet spot keeps rising and thus, for full stride RAID continues to rise
- Files will be striped over a larger percentage of the disks on the floor on average to get desired data rate
- Reliability at scale becomes more and more important



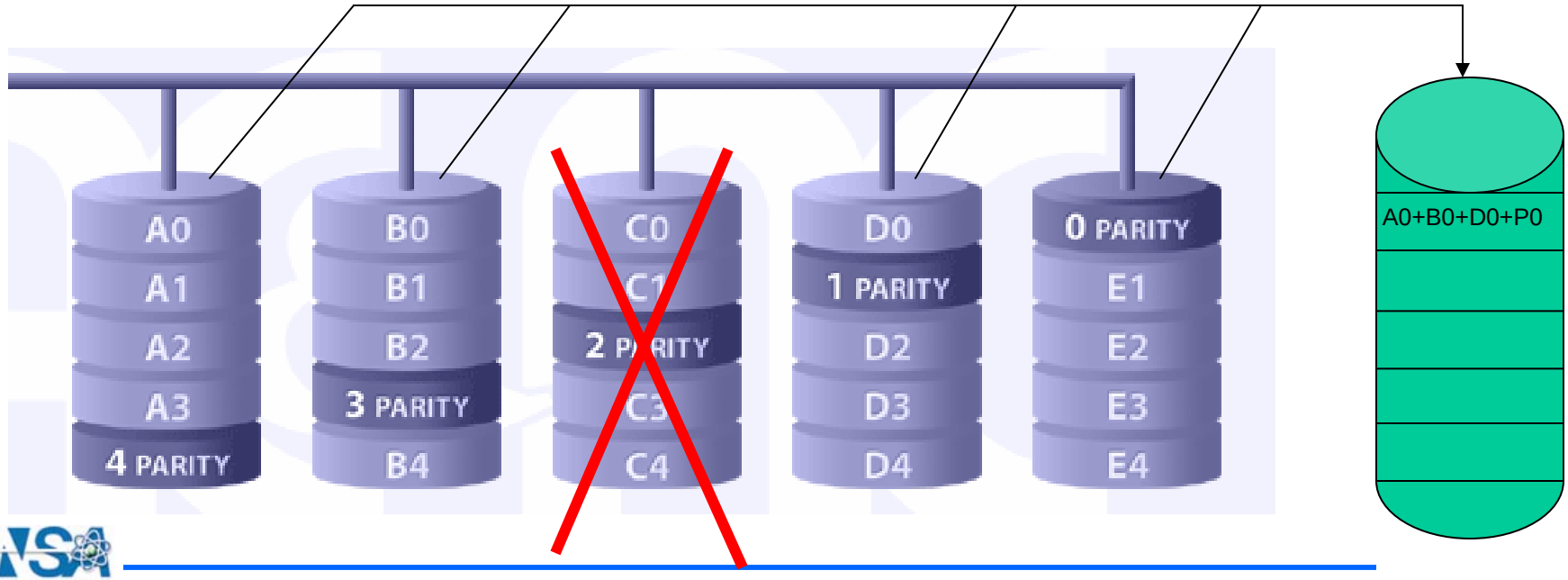
The ASC I/O Ratio and past over engineering of the BW

- ASC ratios (1 GByte/sec per Tflop and 20 Bytes/flop disk)
- In 1996 on a 3 Tflop system, 20 bytes/flop is 60 TBytes of disk, which yealded about 48 GigaBytes/sec which was over engineered by a factor of 16X for BW
- In 2002 on a 20 Tflop system, 20 bytes/flop is 400 Tbyte of disk, which yealded about 40 Gigabytes/sec which was over engineered by a factor of 2X for BW
- Today for a 100 Tflop machine, 20 bytes/flop is 2000 Tbytes of disk yealds a little over 100 Gigabytes/sec, which is not over engineered at all.

- We do not enjoy having far more BW than we really needed to get the space anymore!

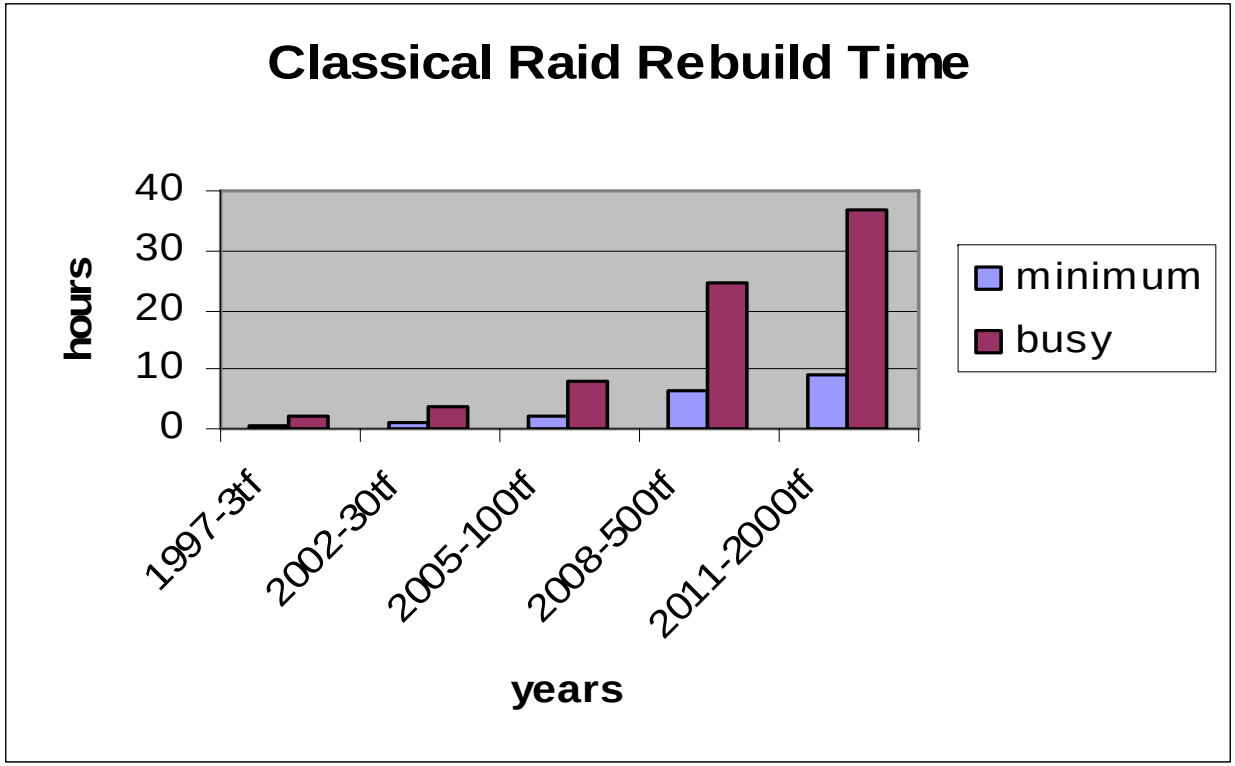
Classical RAID Plus 1 Rebuild

- Read the remaining disks, XOR, and write the result.
- Speed ultimately governed by write speed of target new disk
- This is true for N+1 and N+2 with Classical RAID



Classical RAID Rebuild Time

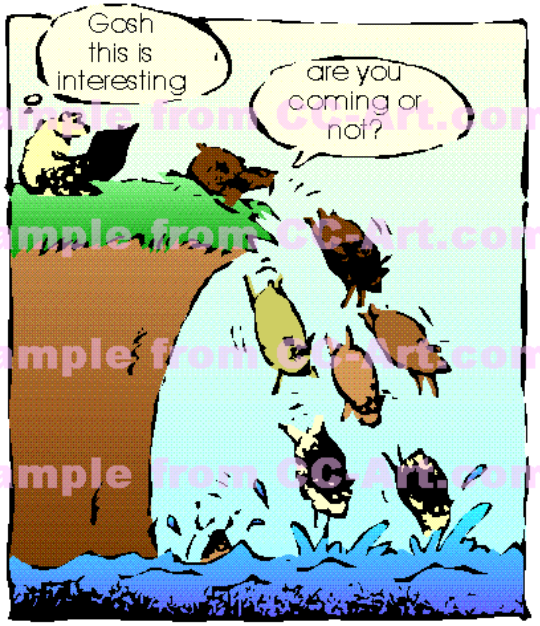
- Rebuild times get worse and worse, from minutes, to hours, to days – raising chances of 2-3 disk failure more and more



The future reliability story

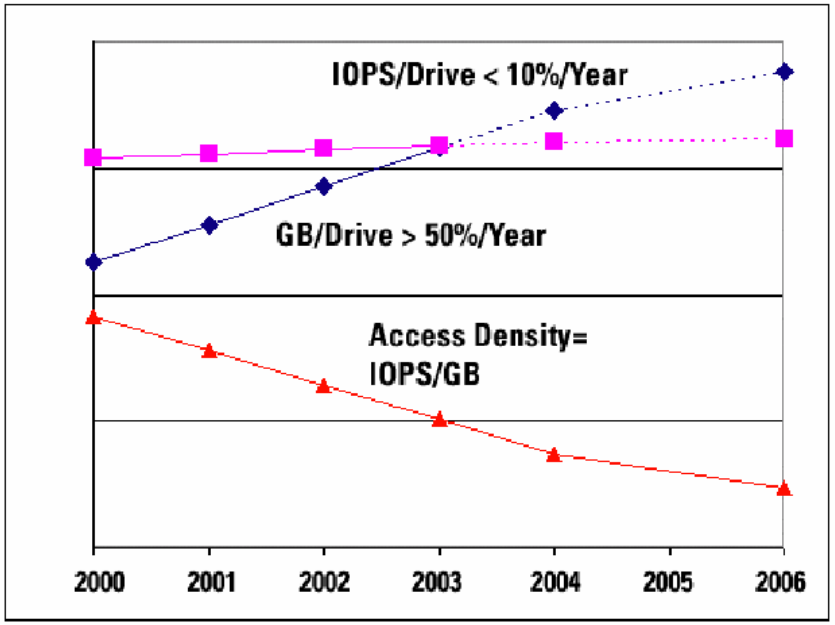
- **We are fighting the combination of**
  - More and more disks to get the job done, driving the reliability down
  - Longer rebuild times driving the reliability down

- **What do we do?**
  - Most solutions are depending on +2 technologies
  - RAID plus 2 technology protects against a two disk loss, but the trend these two issues raises is still less and less reliable over time. You have to collect more data to calculate parity with +2 methods, and this is an important fact!
  - As the collective of machines get larger and less reliable, can we afford to have the mechanism we are using to deal with that growing unreliability (the file system/storage), become less and less reliable?



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.

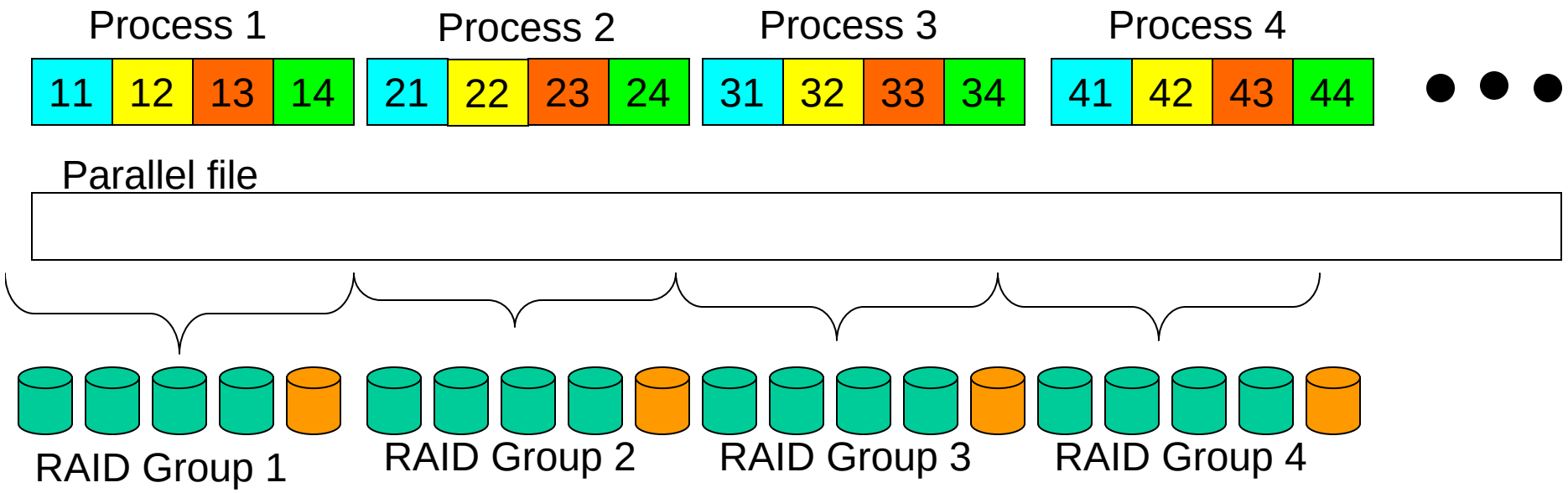


A Disturbing Summary

- ASC ratio driven BW over engineering is no more
- You have to involve more disks to do the job
- Number of disks to get the BW is going through the roof
- Rebuild times get worse and worse
- Plus 2 technologies don't really solve the problem reliability/rebuild problem
- It takes larger and larger write operations to be efficient
- Disks aren't helping us scale metadata either

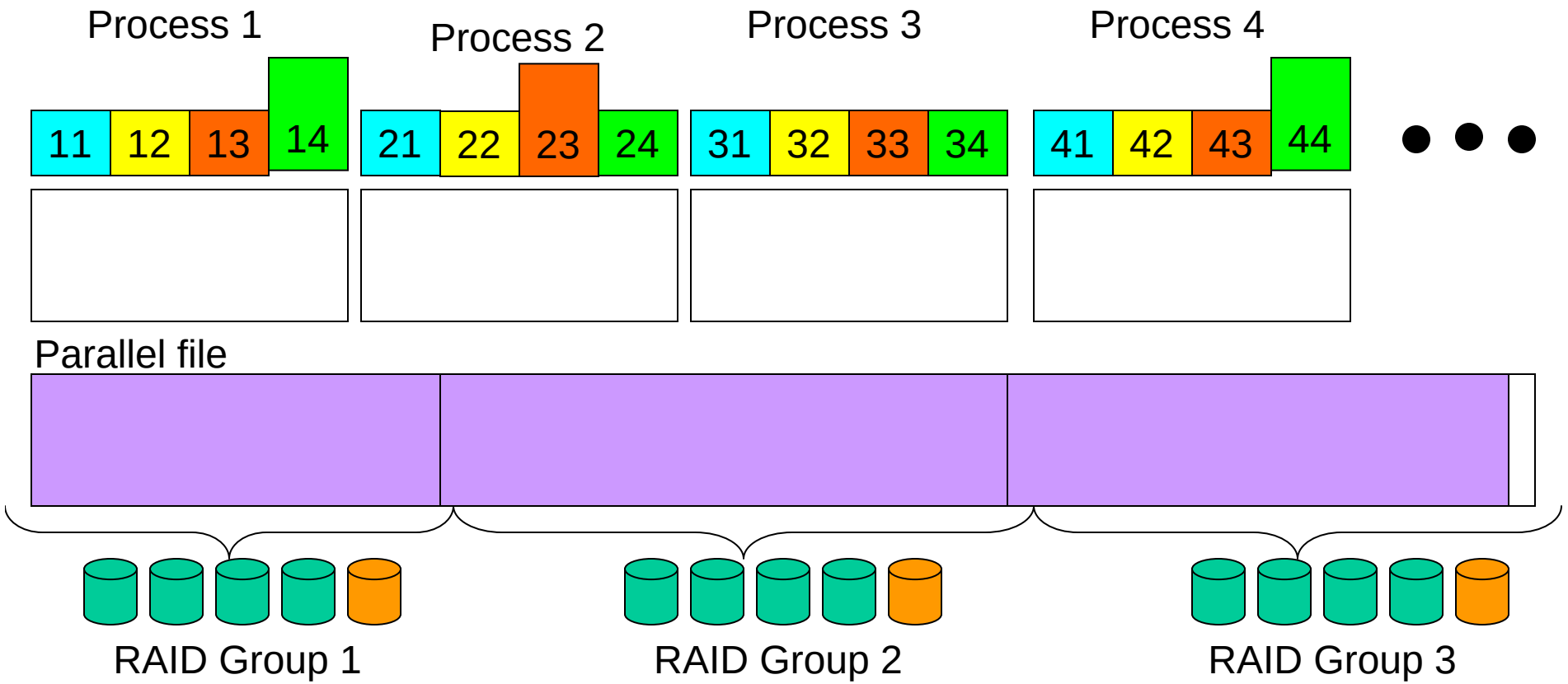
What do apps do?

Example of well aligned I/O



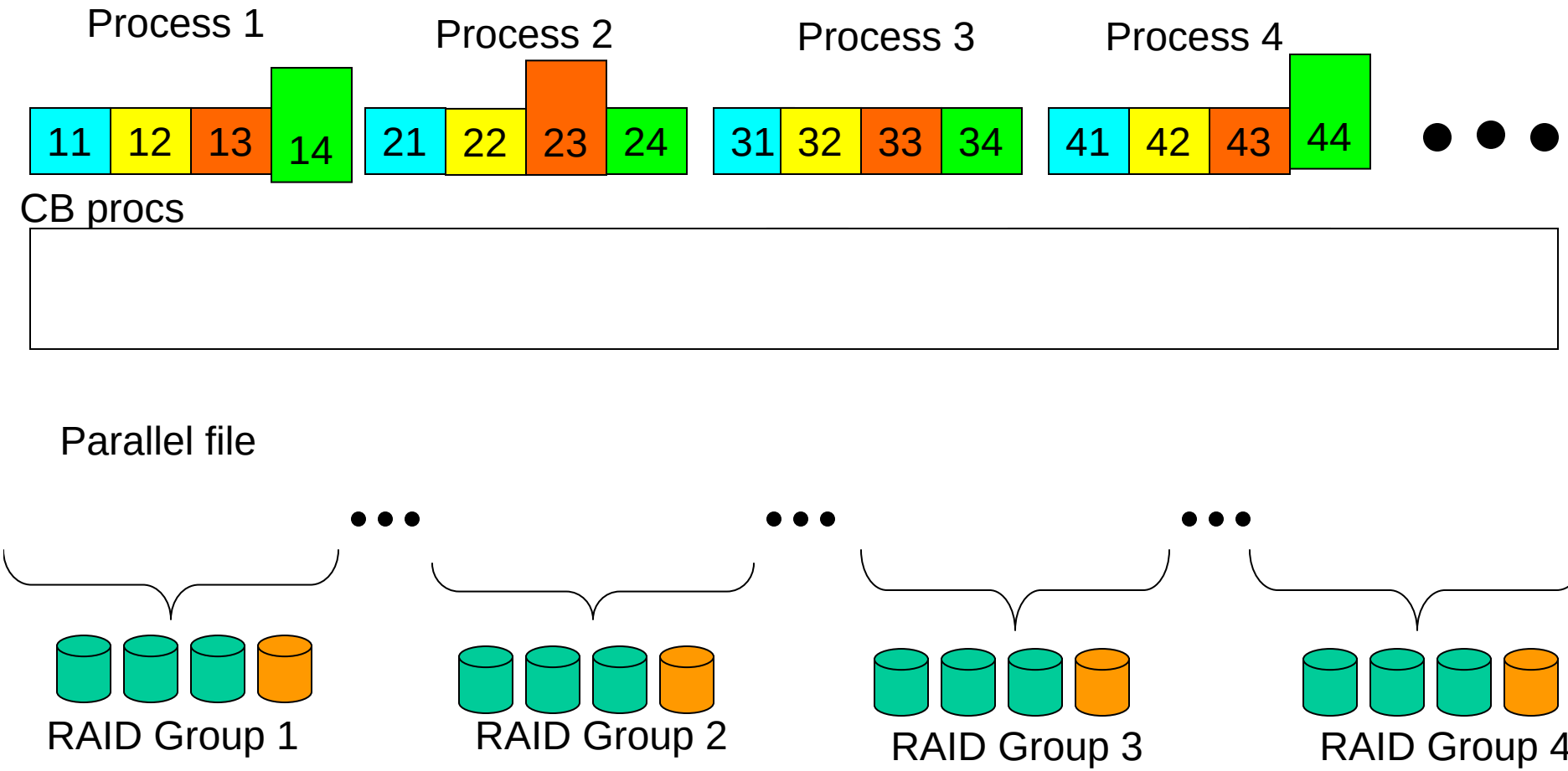
Oh, if applications really did I/O like this!

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



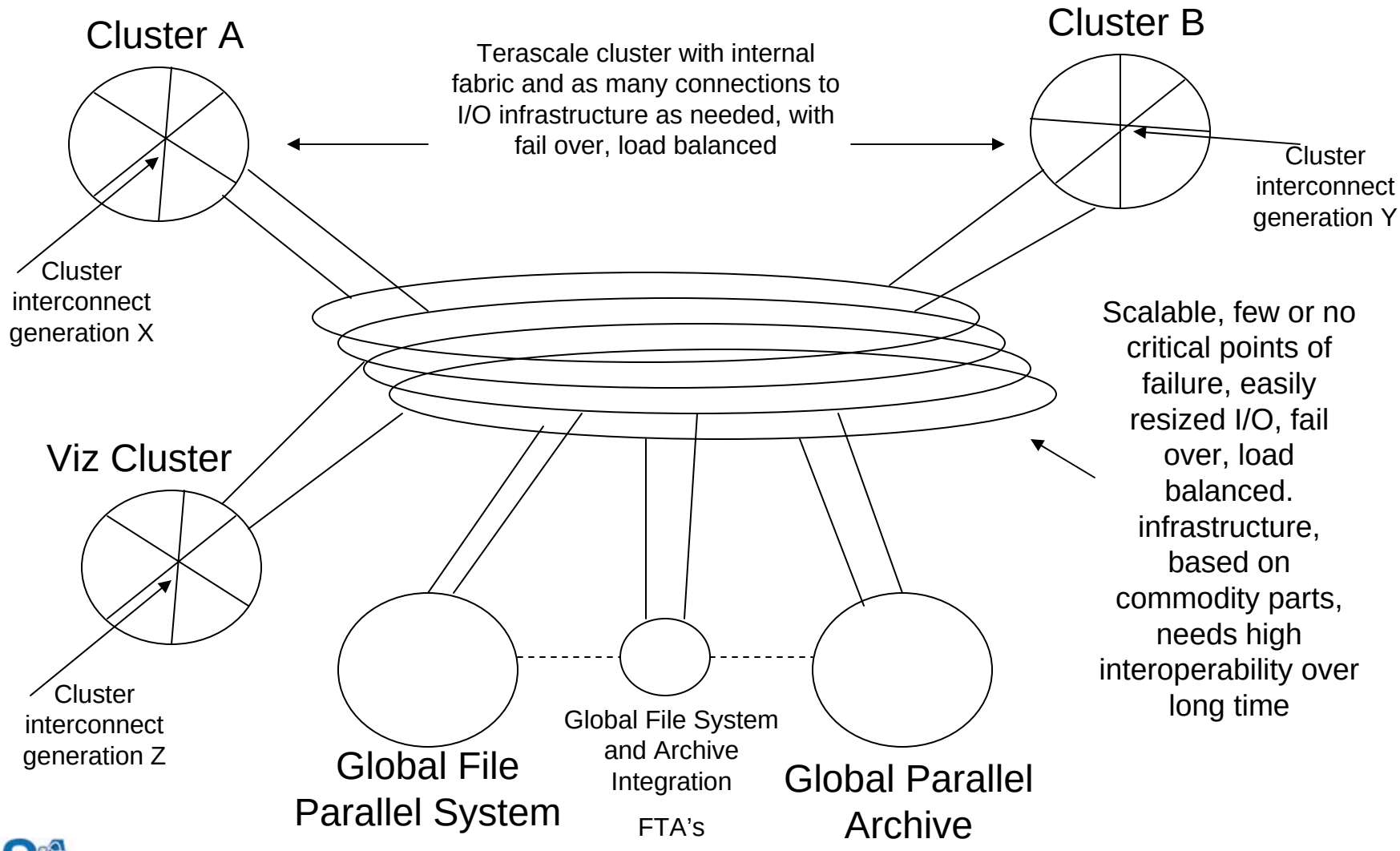
Often, this ends up being an N-squared or N-Log-N problem for the interconnect!

The apps versus the Industry

- CPU's are not getting faster, so we are getting more CPU's.
- Memory per processor is not going up appreciably, in some cases it is going down
- Therefore, apps are not going to write larger writes (and writes are already too small for current storage systems)
- But RAID/Disks are requiring larger and larger write ops for efficiency

What about the Storage Network?

Global connection of multiple terascale clusters to a common file system



Well, maybe it is harder than it seems and getting harder by the month?

Of course we have come a long way since the mid 1990's with parallel file systems and I/O stacks. We also have made some great strides in spinning up R&D in this area as well.

Hopefully you will hear about some interesting approaches to these problems and others this week. If you don't and you still want to know more, just ask your local I/O Nerd, or catch me in the hallway.