The ASCI/DOD Scalable I/O History and Strategy

Run Time Systems and Scalable I/O Team
Gary Grider
CCN-8
Los Alamos National Laboratory
LAUR 042787

05/2004

Parallel File Systems and Parallel I/O

Why - From the ASCI implementation plan:

- "Responsible for ensuring that ASCI applications have access to reliable, easy-to-use, high-performance input/output systems whose throughput rates and capacity are in balance with the computational speed and memory size of the ASCI platforms…"
- "Provide standard parallel I/O interface to all ASCI Tri-Labs applications and throughout the HPC community – MPI-IO"

How – Leverage, Partnerships, Careful Planning, and did I mention Leverage?

- Vendors
- Standards bodies
- Universities
- HPC Community

In 1995-1996 " Oh no, we don't have a good have a good scalable file system story"

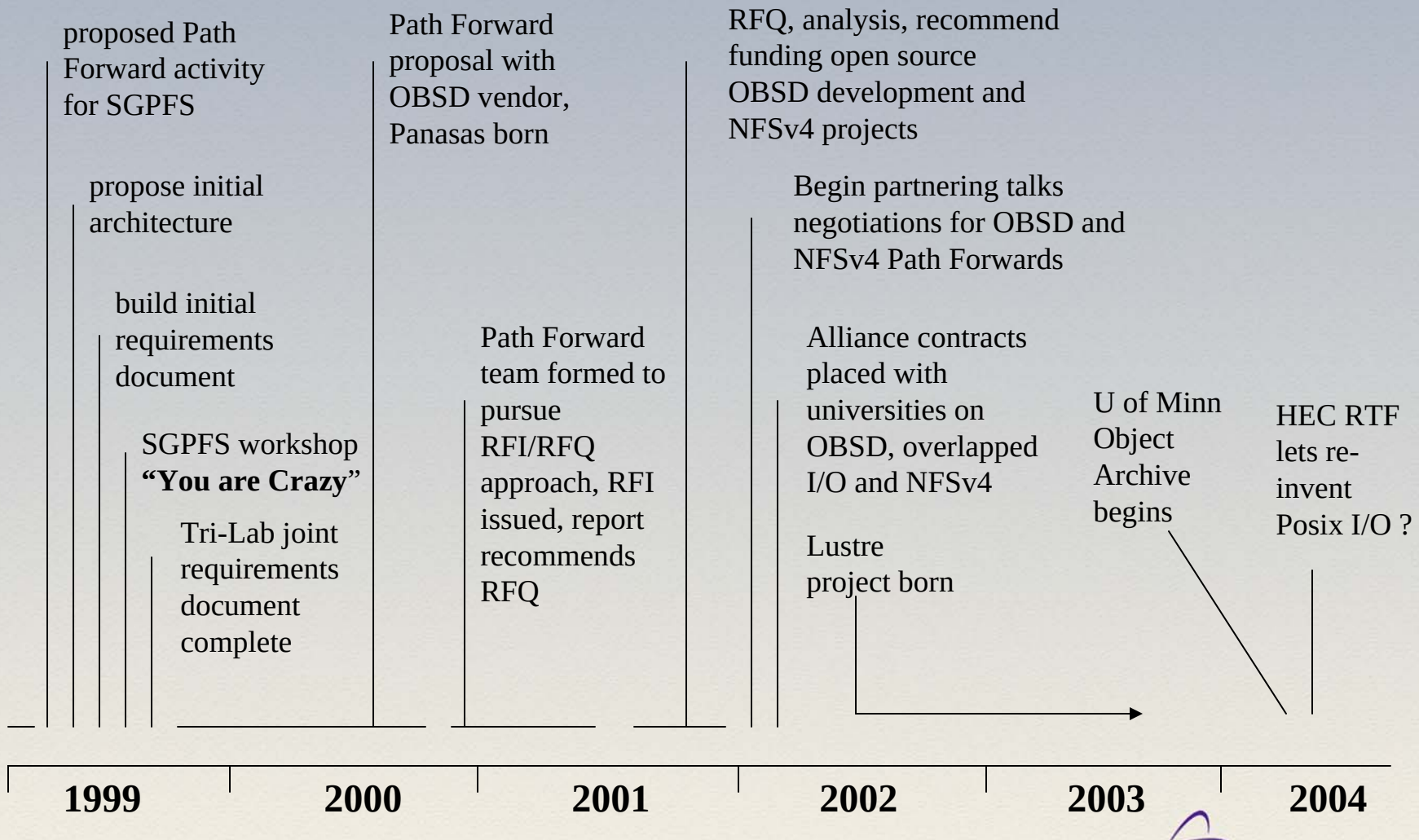
For Sandia, LLNL, LANL and DOD, the need for a global parallel file system was there from the beginning of clustered based parallel computing,

- **few solutions existed,**
- **none were heterogeneous,**
- **none were open source,**
- **none were based on standards, and**
- **none were secure on a public net.**

This is primarily for our giant clusters, secondarily for our enterprise, and lastly across multiple enterprises/sites

We saw Linux clusters coming in the future which made the problem very real and very evident

Historical Time Line for R&D



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 1

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***

FS Requirements Detail 2

4.4 Scalable Management & Operational Facilities

- ***4.4.1 Need to minimize human management effort***
- ***4.4.2 Integration with other Management Tools***
- ***4.4.3 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***

FS Requirements Detail 3

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***

It Has to Scale with Our Machine Appetite

Aggregate Bandwidth Rates for One Parallel Job
Simulation & Physics Model Aggregate FS Requirements

	1999	2003	2005	2008
Teraflops/Clients	3.9 / 6K	30 / 12k	100 / 50K	400 / 100k
Memory Size (TB)	2.6	13-20	32-67	44-167
I/O Rates (GB/s) N to N and N to 1	4 – 8	20-60	50-200	80-500

Other things have to Scale Too

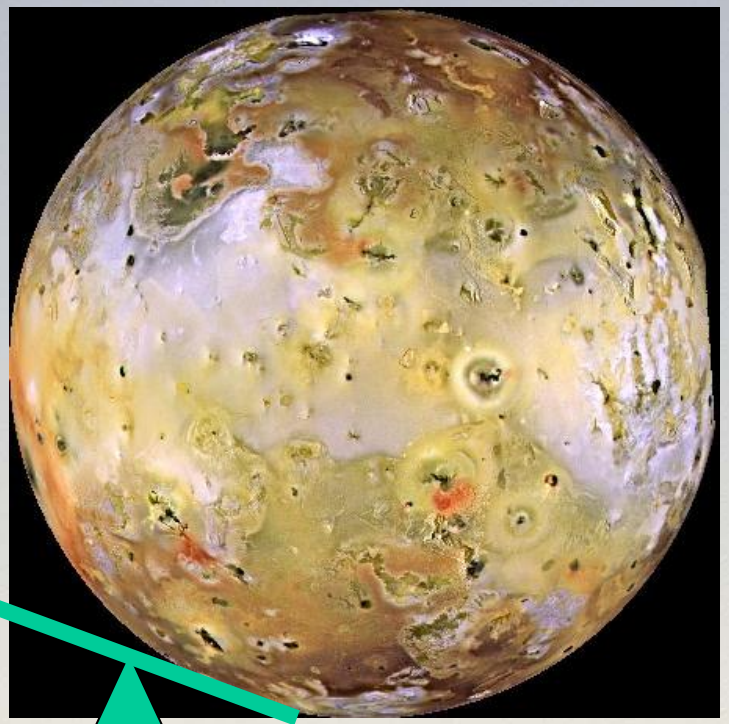
File System Attributes				
	1999	2002	2005	2008
Teraflops	3.9	30	100	400
Memory size (TB)	2.6	13-20	32-67	44-167
File system size (TB)	75	200 - 600	500 -2,000	20,000
NNumber of Client Tasks	8192	16384	32768	65536
Number of Users	1,000	4,000	6,000	10,00
NNumber of Directories	5.0*10^6	1.5*10^7	1.8*10^7	1.8*10^7
NMetadata rates (random full stat or completed insert)	500/sec 1 mds	2000/sec 1 mds	20,000/sec n mds	50,000/sec n mds
NNumber of Files	1.0*10^9	4.0*10^9	1.0*10^10	1.0*10^10

Other Requirements Besides Scalability

- Based on Standards (ANSI T10 is now acceptance of Draft 1)
- Security more like AFS/DFS but better
  - 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 (ANSI T10), (NFSv4.X IETF), (formal POSIX enhancements), all in progress
- POSIX behavior with switches to defeat parts
  - 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, only glimpses of products now, and already nearing 100 sites trying out one of the 3-4 solutions, expected to be in the 1000's in 1-2 years.

I promised you Leverage, Collaborations, Planning, and Leverage

Vendors,
other labs,
universities,
standards, etc.



Vendor Collaborations

Solution for Linux clusters and enterprise class heterogeneous global parallel file systems

- HP/CFS/Intel Lustre Path Forward for object based secure global parallel file system
  - Very scalable bandwidth, good non scaled metadata
  - Being used at LLNL
  - Starting metadata scaling work
- Panasas
  - Very scalable bandwidth, good non scaled metadata
  - Being used on viz clusters in open, deployed on 1000+ cluster in open and 1400+ cluster in the secure

Vendor Collaborations (Continued)

MSTI's MPI-IO

- Advanced features
- ADIO layers for Panasas and Lustre

IBM's Storage Tank (collaborations just beginning)

- Beginning to work with IBM-Ohio Supercomputer Center for bandwidth scaling etc.
- HSM design for Object file systems
  - Tied to our University of Minnesota Intelligent Storage Consortia work
- NFSv4.X as native client for SAN and Object file systems
  - Tied to our University of Michigan NFSv4 work, In IETF now!
- End to End secure file systems

Additionally, Our University Partnerships

Michigan

- Assisting in design and testing of NFSv4 for multi-domain enabled secure NFS, NFS parallel extensions, and NFS in a non IP environment, NFS as a client to SAN and OBFS file systems, NFS server load balancing in a cluster file system setting
- Some results are showing up in Linux 2.5/2.6 kernel, pNFS IETF work starting

Northwestern

- Move coordination up to MPI-IO, out of the file system for overlapped I/O

UCSC

- Study on object storage efficiency and clustered metadata scaling, to guide our two object file system future design and development activities

Minnesota (Intelligent Storage Consortia) and STK to develop first infinite (HSM) object based device to enable parallel object archive

- Leverage existing commercial HSM software (multiple copies in parallel)

Collaborations with other labs

ANL

- MPICH MPI-2 reference
- MPI-IO advanced features and ADIO
- PVFS2 explorations
- NetCDF
- GridFTP

PNNL

- Lustre

Brookhaven

- Panasas

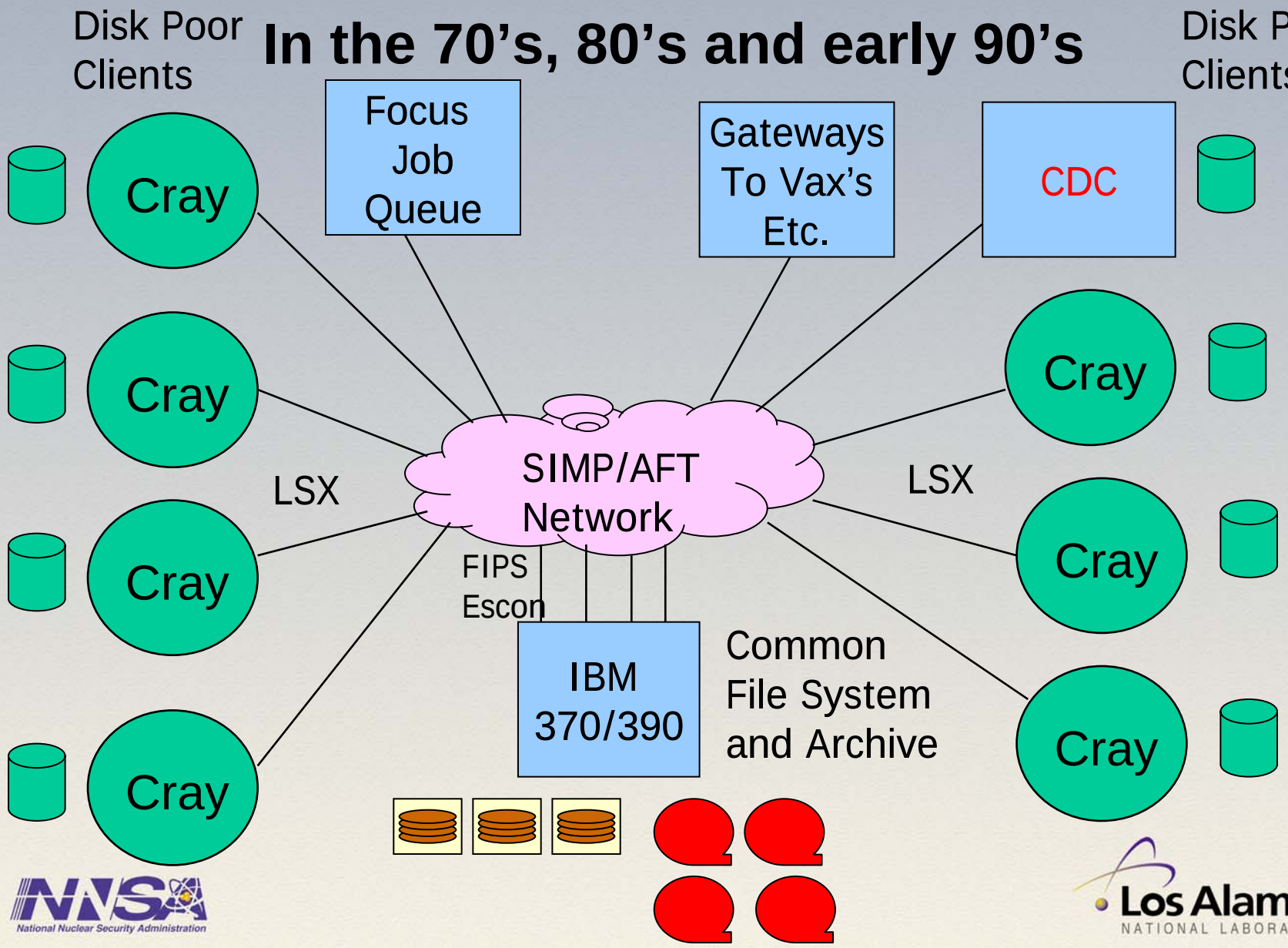
OSC

- Storage Tank as parallel file system

High End Computing Revitalization Task Force Identified R&D Targets

- Lee Ward of Sandia, Tyce McClarty of LLNL, and I were the I/O reps at HEC
- Overwhelming idea in the I/O area was
  - If we get all these scalable and parallel file systems, and if we get devices that are smart, and if we want to extend the idea of PIM all the way to the storage devices, and if we have a secure way to ask devices to work for us, and if we have interesting OBSD's, and if we get NFS to be a good file system client for SAN and OBFS, etc.
  - How can we possibly utilize such things if we are forced to go through an the current POSIX (open, read, seek, write, close)?
  - Furthermore, are trees the way to organize a baZILLION files? Is readdir and stat going to be able to service us much longer?
- So, should we create a new I/O api legacy on which to build for the next several decades?
- We are spinning up a "enhance Posix" effort as we speak!

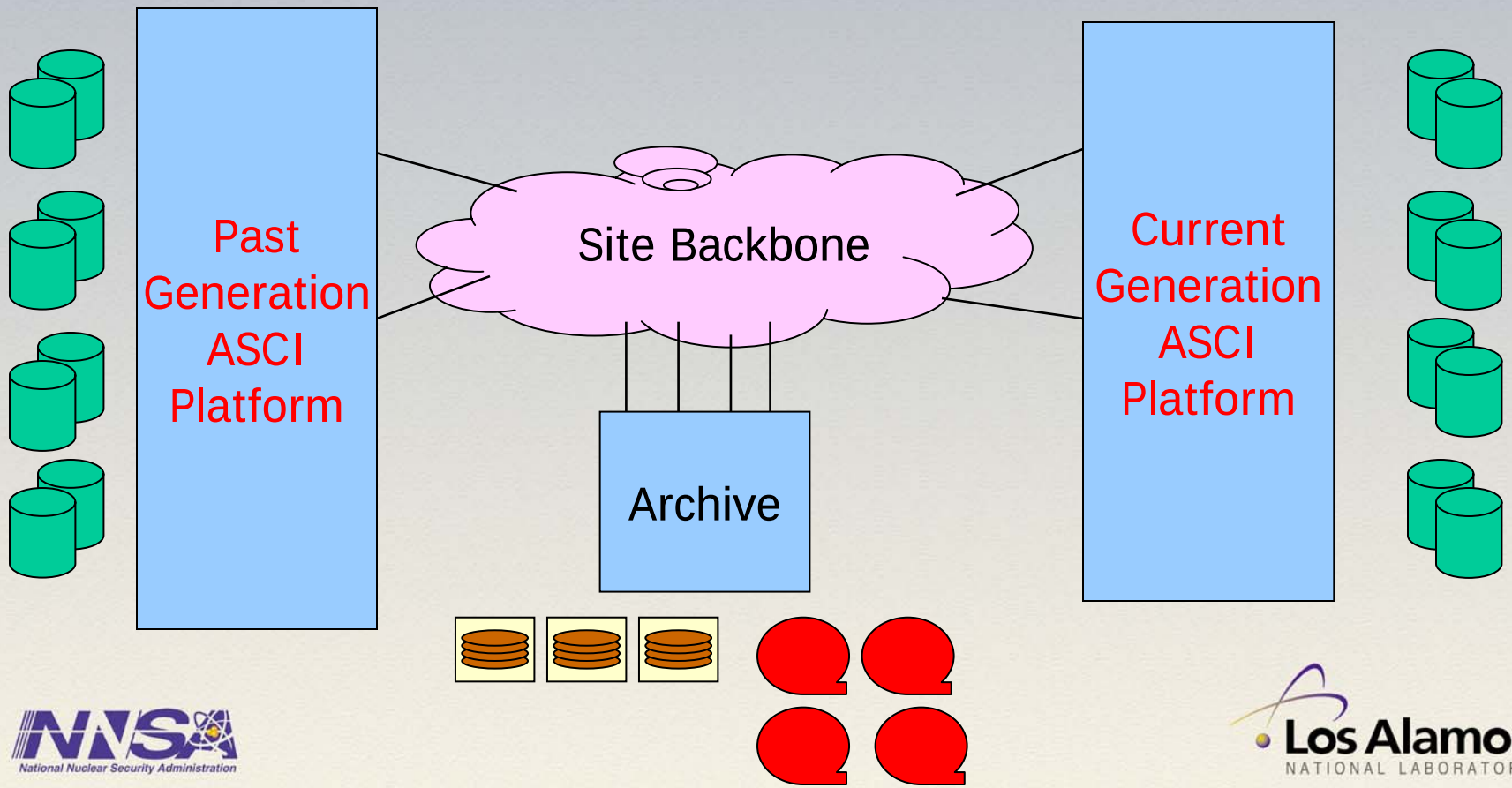
In the 70's, 80's and early 90's



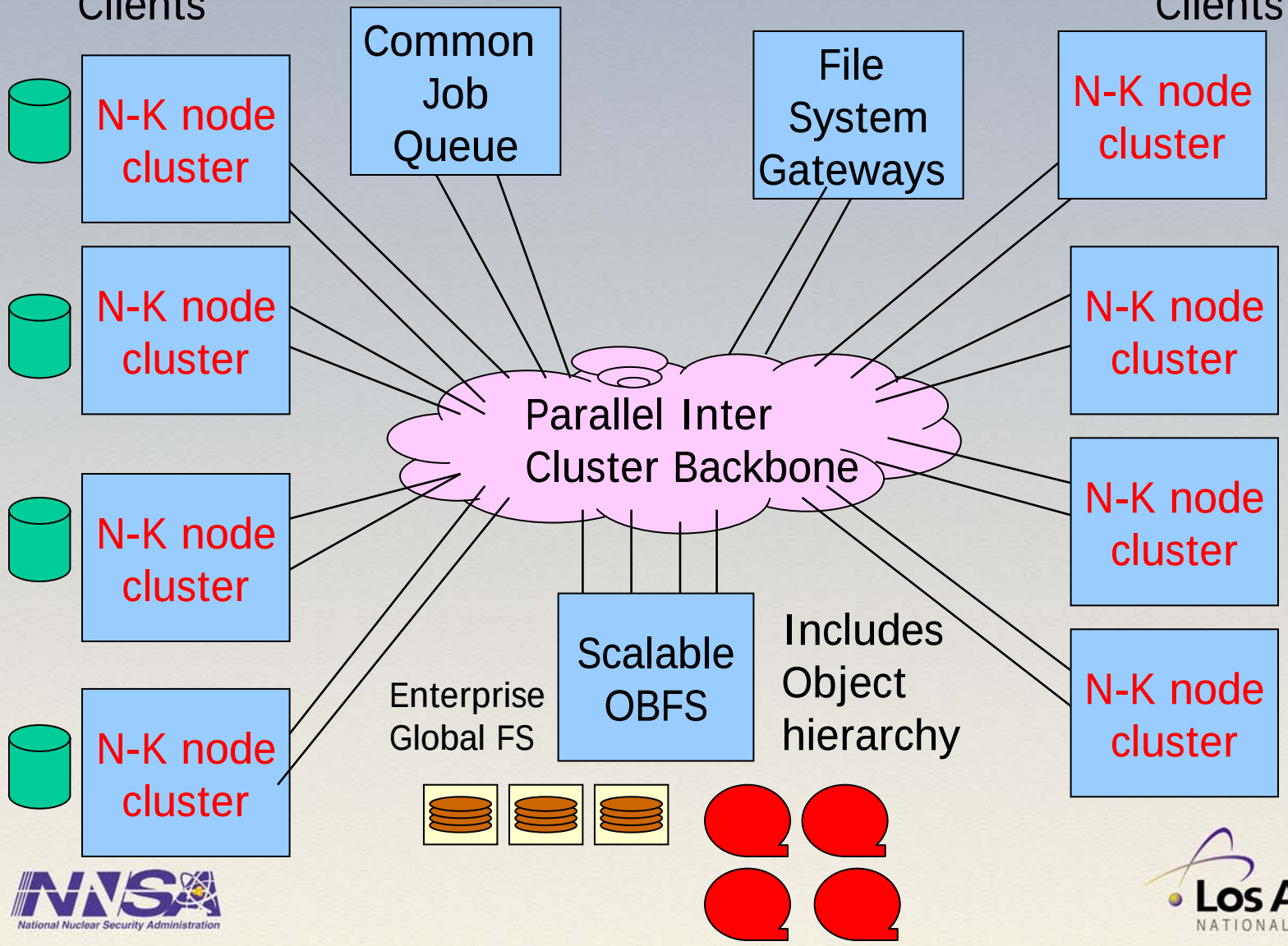
In the Late 90's with ASCI

Disk Rich
Supercomputer
With its own
parallel file system

Disk Rich
Supercomputer
With its own
parallel file system



In the 21st century



Summary

We have been at this SIO game for a while

We have been building strong leveragable partnerships

We have helped shape standards (T10, NFS4, POSIX)

Linux and Enterprise class secure global parallel file systems being deployed now and growing

- Lustre in use, moving into programmatic role
- Panasas in use, moving into programmatic role
- Building relationship with Storage Tank
- Headed towards enterprise class common parallel file systems

What more can we do?

Assumptions

- an OSD standard becomes a reality (so we have a secure infrastructure to ask storage devices to do things for us) – Wow T10 is out!
- we get one open source scalable global parallel capable file system based on OSD out there in use – Wow Lustre is out there
- a couple more scalable global parallel capable file systems based or headed towards OSD out there – Wow Panasas and Storage Tank are out there
- an NFSv4.X implementation in Linux distribution that has agnostic way to be an OSD file system client – Wow, the IETF pNFS effort is cranking up
- other non Linux OS vendors follow with NFSv4.X clients – Wow, Solaris, AIX, Netapp, HP, etc. are at least doing V4 without pNFS support, and pNFS has strong support with these folks and more.

Not a bad set of vectors pointed in the right direction!

What more can we do? Well, that is why I am here!

What do we do next?

- **change POSIX to help HPC apps (lose the stream of bytes ordering)**
- **fund archive objects (U of Minnesota work to investigate parallel object archive leveraging everything so far)**
- **fund (next FY) research for function offload, PIM all the way to the storage, how to we further exploit Ojbect Device Intelligence?**
  - **DOE/NNSA move parts of UDM/HDF/NetCDF etc. to devices**
  - **DOD information from data**
  - **Databases, filtering, real-time, etc.**

  can we deal with QOS, security, and other issues?

  how do we change POSIX to allow for function offload?