System input-output, performance aspects

March 2009

Guy Chesnot

Agenda

- Data sharing Evolution & current tendencies
- Performances: obstacles
- Performances: some results and good practices

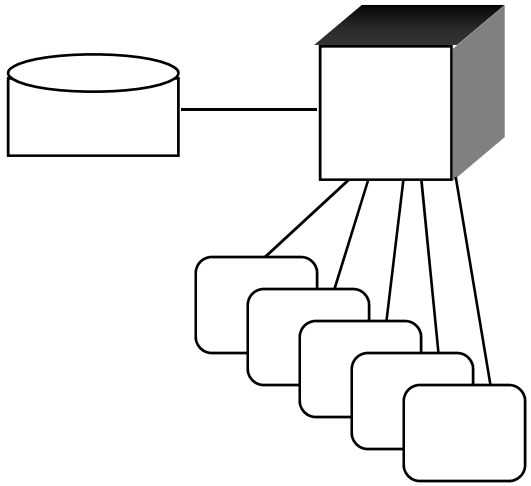
A file system taxonomy

- Local File System - The generic term for a non-shared file system
  - Examples: XFS, EXT2, EXT3, NTFS, FAT, …
- Distributed File System - The generic term for a client/server or "network" file system where the data is not locally attached to a host.
  - Network File System (NFS) is the most common distributed file system currently in use for Open Systems.
- Storage Area Network (SAN) File System – Provides a means for hosts to share Fiber Channel storage.
  - Examples include: CXFS, IBM's General Parallel File System (GPFS) and Quantum's StorNext File System
- Parallel File System – Meaning a transport by many servers, towards many clients.
  - Examples: pNFS (in the future), Lustre (SUN), Panasas (Panasas)

Agenda

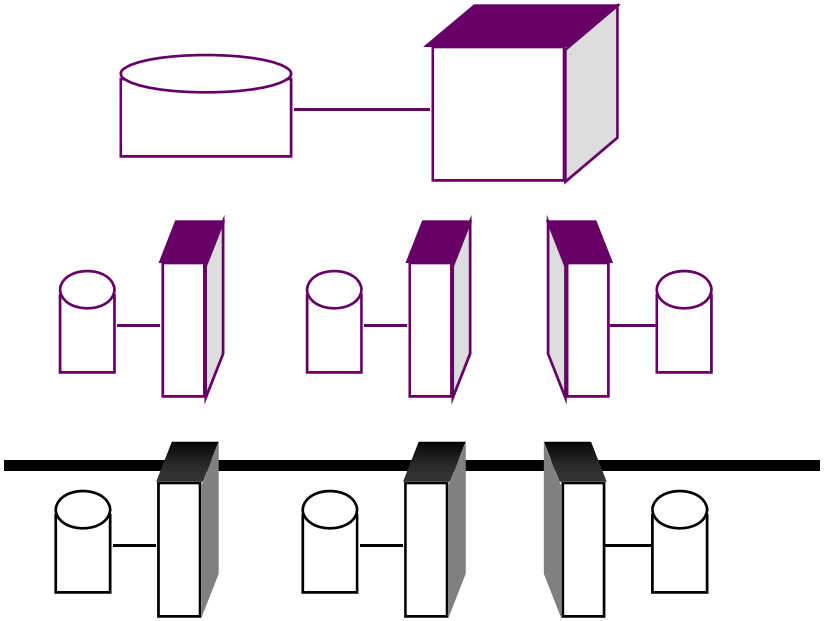
- **Data sharing Evolution & current tendencies**
  - **Good old times !**
  - **Ups & downs**
  - **File systems: local, distributed, shared, parallel**
- Performances: obstacles
- Performances: some results and good practices

Good old times ?



Mainframe: centralized computing

No network
All data is shared



Groups of workstations

Local network(s)
Some data is shared

Ups: Performance

- In the old way, peak performance was related to a limited number of disks
  - New obstacles appeared
  - Operating system (kernel + local file system manager)
  - Hosts servers

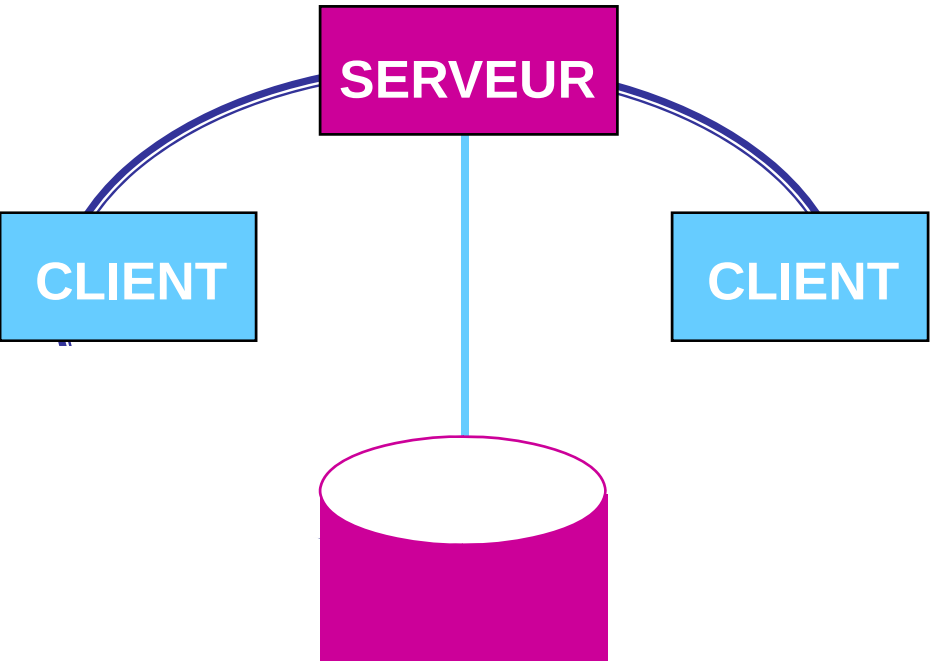
- Aggregate I/O requests (by our customers)
  - In the old times, Linux 2.4 kernel, below 300 MB/s
  - Current times, Linux 2.6 kernel, over 5 GB/s

Ups

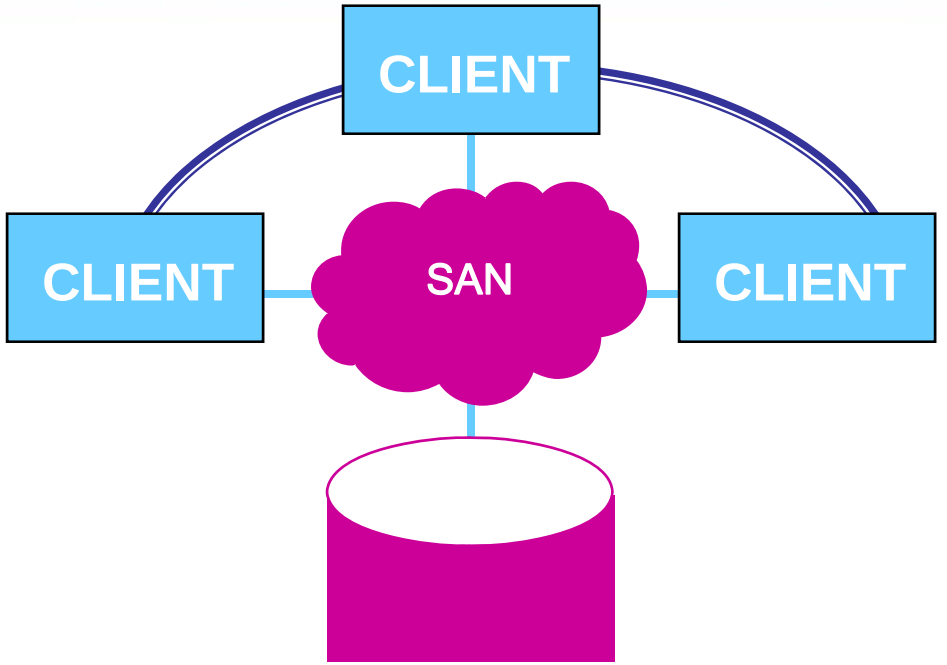
- Parallelism on servers and PCs: multi cores processors
- Memory size
  - Faster than disk transfer rate
  - > more disks pour to fill memory at the same rate
- Disk unit capacity
  - Faster than disk transfer rate
  - > less disks for the same data capacity
  - > need to grow the aggregate dik unit transfer rate
- Volume of data (downfall*)

* Naming by IN2P3

From distributed to SAN file systems



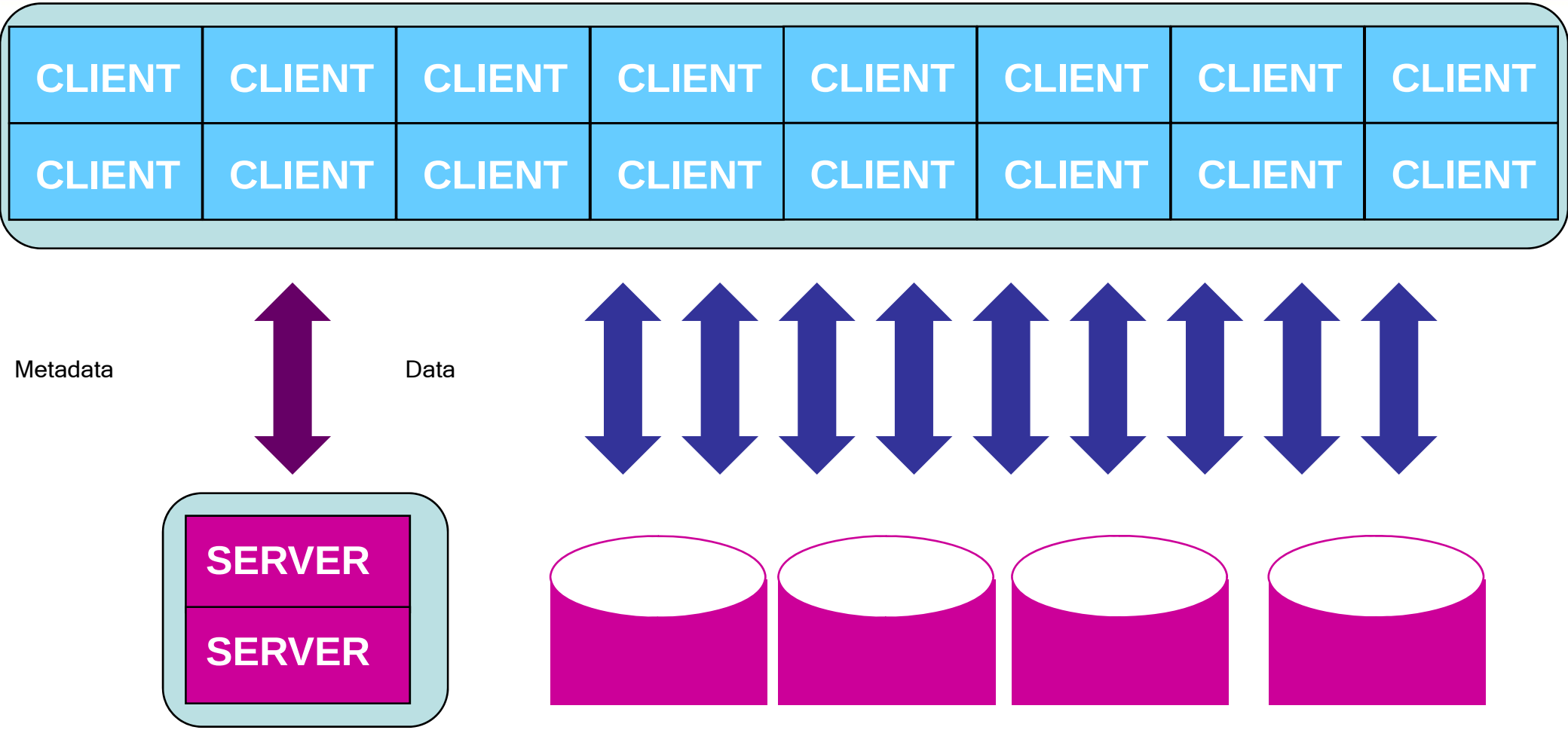
Distributed file system



Symmetric shared file system

Each client is in charge of consistency check.

Parallel file system



Lustre, Panasas: a common design

- Objects
  - Data stored as object
  - Metadata server handles objects instead of blocks
  - Global namespace
- Parallelism, data access through high performance LANs
  - => goal is linear scalability for capacity and performance
- Centralized Architecture : for one data center
  - = one file system + one architecture
- Redundancies to ensure availability in case of hardware failures
- Better use own client-server protocol

Lustre, Panasas: some differences

- Lustre: Open Source model
  - Software only
  - Hardware (storage, servers) agnostic
- Panasas: Appliance model
  - Cluster of metadata servers (metadata scalability)
  - Cluster also for storage (data scalability)
  - Built-in redundancies
  - Integration of protocols (network, storage)
- Different networks of choice
- What kind of managed clients in the future?

Agenda

- Data sharing Evolution & current tendencies
- **Performances: obstacles**
- Performances: some results and good practices

Numbers of clients

- Tens
  - Distributed, SAN, Parallel file systems
- Hundreds
  - Distributed, SAN (some), Parallel file systems
- Thousands
  - Only Parallel file systems can reach these figures

What kind of clients ?

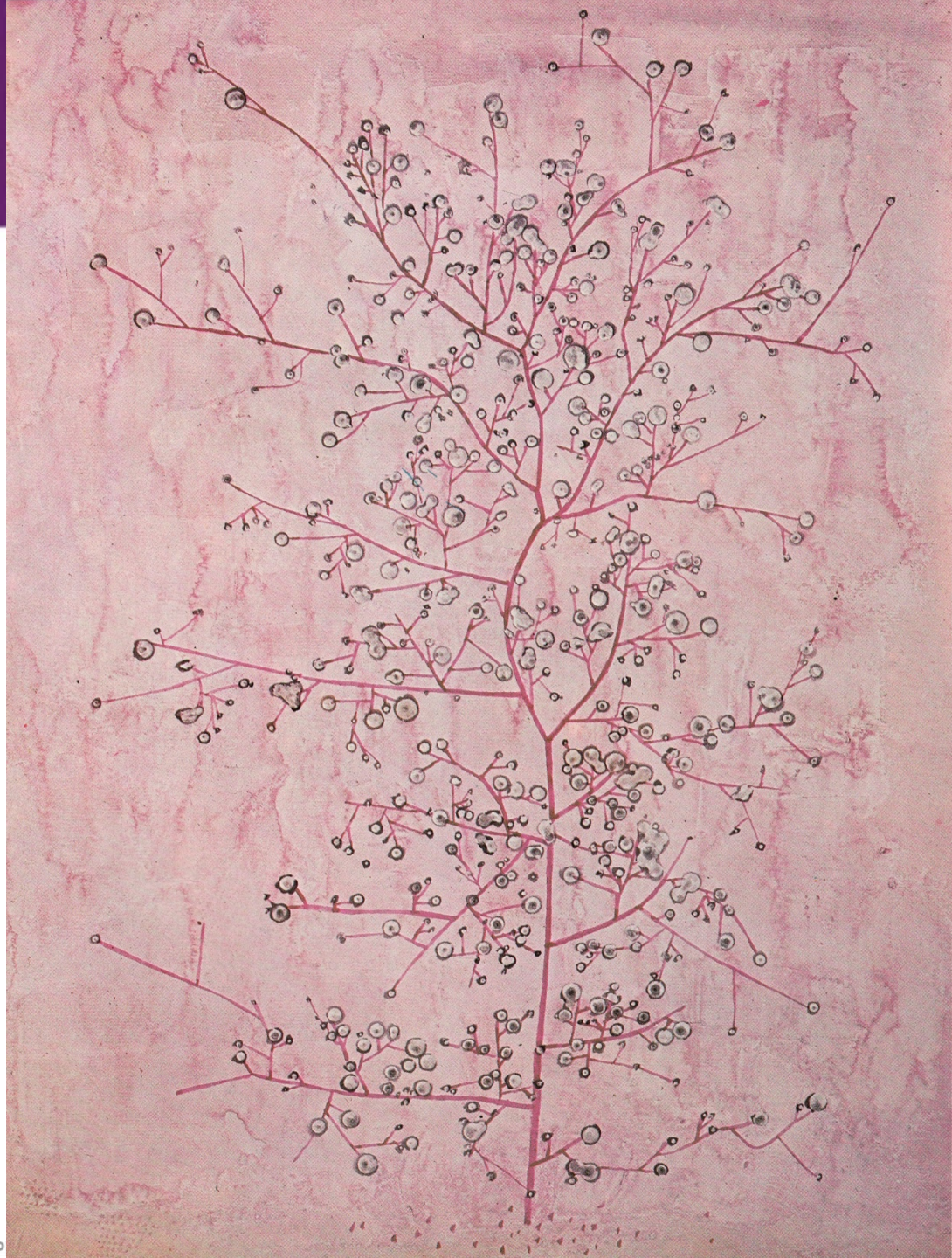
- Distributed (NAS: NFS, CIFS): universal access
- SAN file systems; depending upon architectures
  - GPFS: limited
  - CXFS: large span (UNIXs, LINUXs, Windows's, Mac)
- Parallel file systems
  - Limited
  - Waiting for pNFS (Godot ?)

Number of files

- Scanning hierarchy tree with numerous files

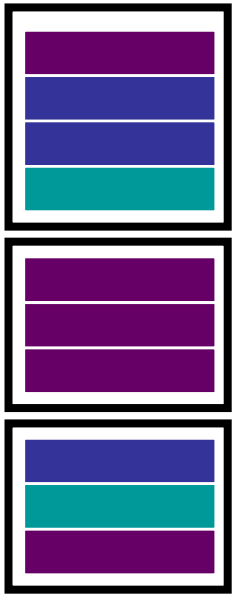
- **Solutions**
- Same external interface
- Different internal architecture
  - XFS local file system

- Objects



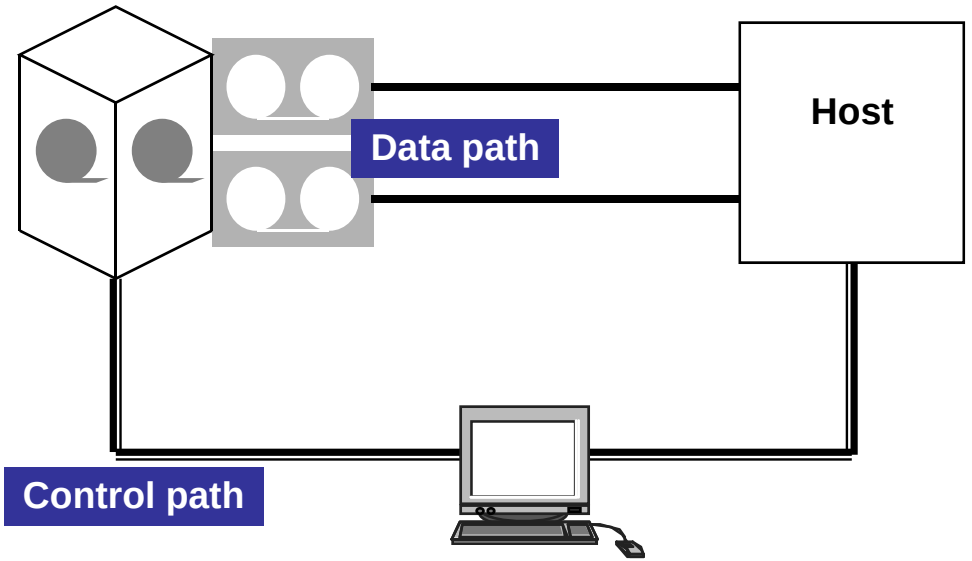
Variable file size

- Usual data handling : blocs
- Object is now more popular
  - Hardware level: more “clever” disks
  - Software level: pNFS, Lustre, Panasas



Links congestion

- Data and metadata on same link
  - Cars and bicycles on the highway
- **Solution**
- Third party transfer, also called out-of-band or split-path
  - Different paths for metadata (requests, status) and data
  - Server initiated data transfer
  - Direct transfer between host and storage
  - Example: Tape library



Disk technology

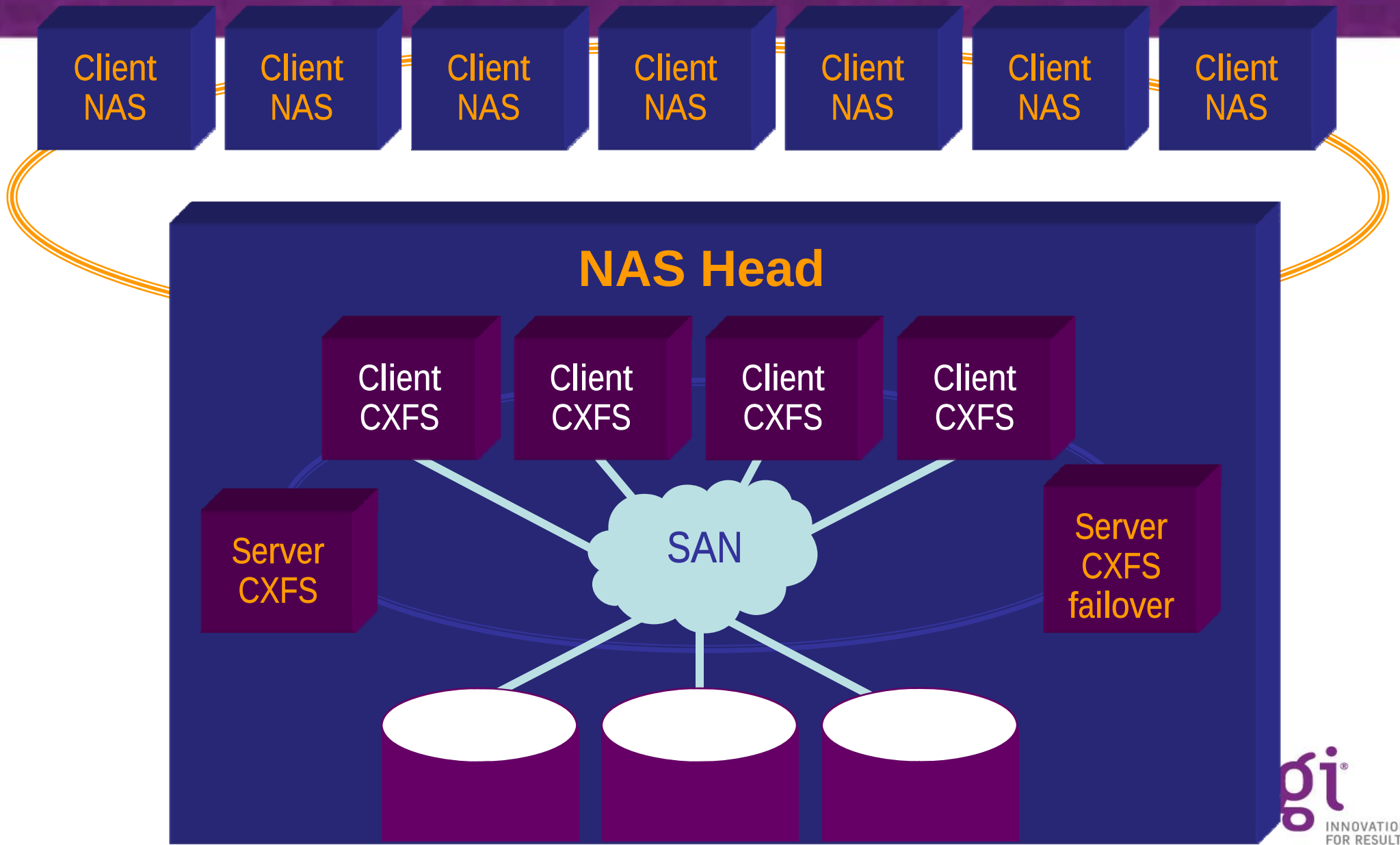
- Better
  - reliability, form factor decrease
  - Cost per Gigabyte
  - Density, doubling every three years (average)
- Hardly better
  - Transfer rate
  - Access time (30% in 10 years)
- => lesser than CPU power
- **Solution**
- Parallelize the access
  - Parallel file system

Performance scalability limit

- **Distributed file system**
  - One server only
  - Data and metadata management on same host
  - > limited number of clients

- **Solutions**
- Clustered NAS = SAN file system
- Parallel file system

Clustered NAS at SGI



Performance scalability limit (cont. 1)

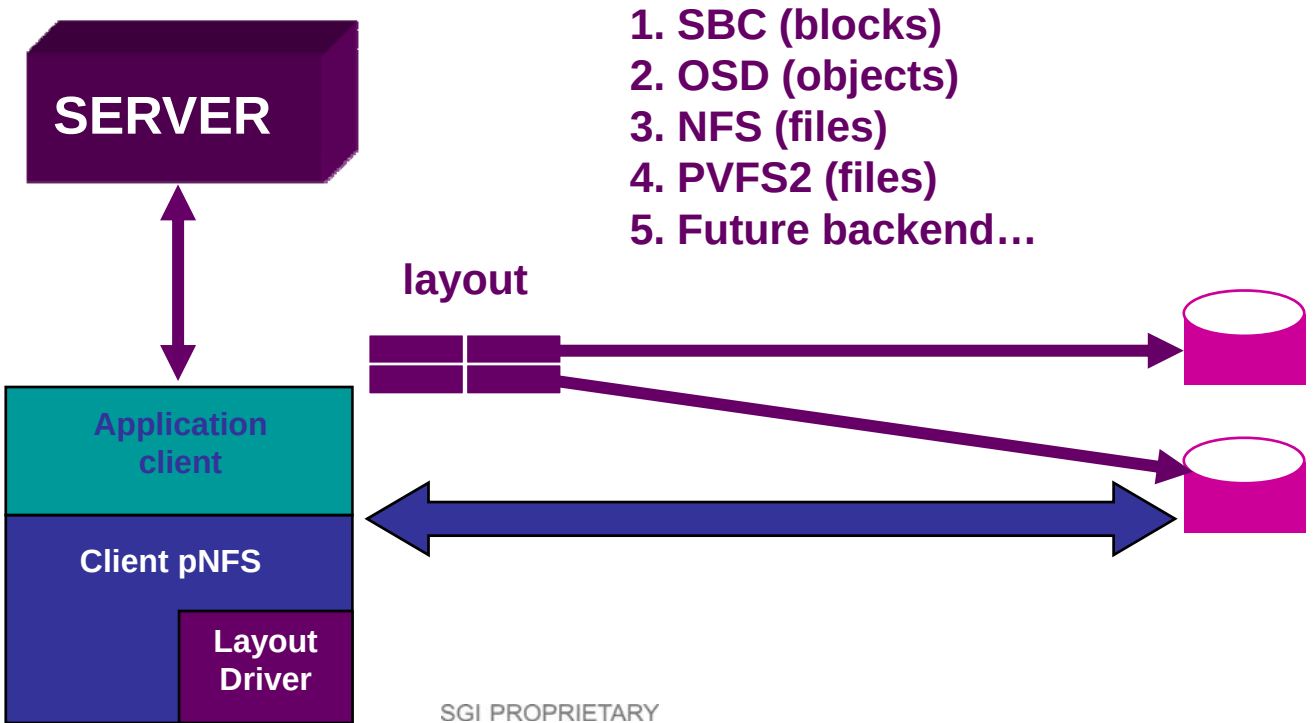
- **SAN file system**
  - No user mobility
    - On one SAN
    - On a limited number of OS
  - Asymmetric: limited number of metadata servers
    - Wide client heterogeneous Os's
  - Symmetric: more metadata servers
    - Few client heterogeneous Os's
    - Not so good for export via NFS towards light clients
    - Data and metadata management on same hosts
- **Solution**
- Numerous SAN, integrated by a parallel file system

Performance scalability limit (cont. 2)

- **Parallel file system**
- Metadata parallelism
  - **Solution** : possible, product architecture is not an obstacle
- Cache consistency improvement
  - **Solution** : nothing new in pNFS until now
- Higher complexity pNFS
  - Layouts management
  - **Solution**: Lustre, Panasas (one kind of layout)

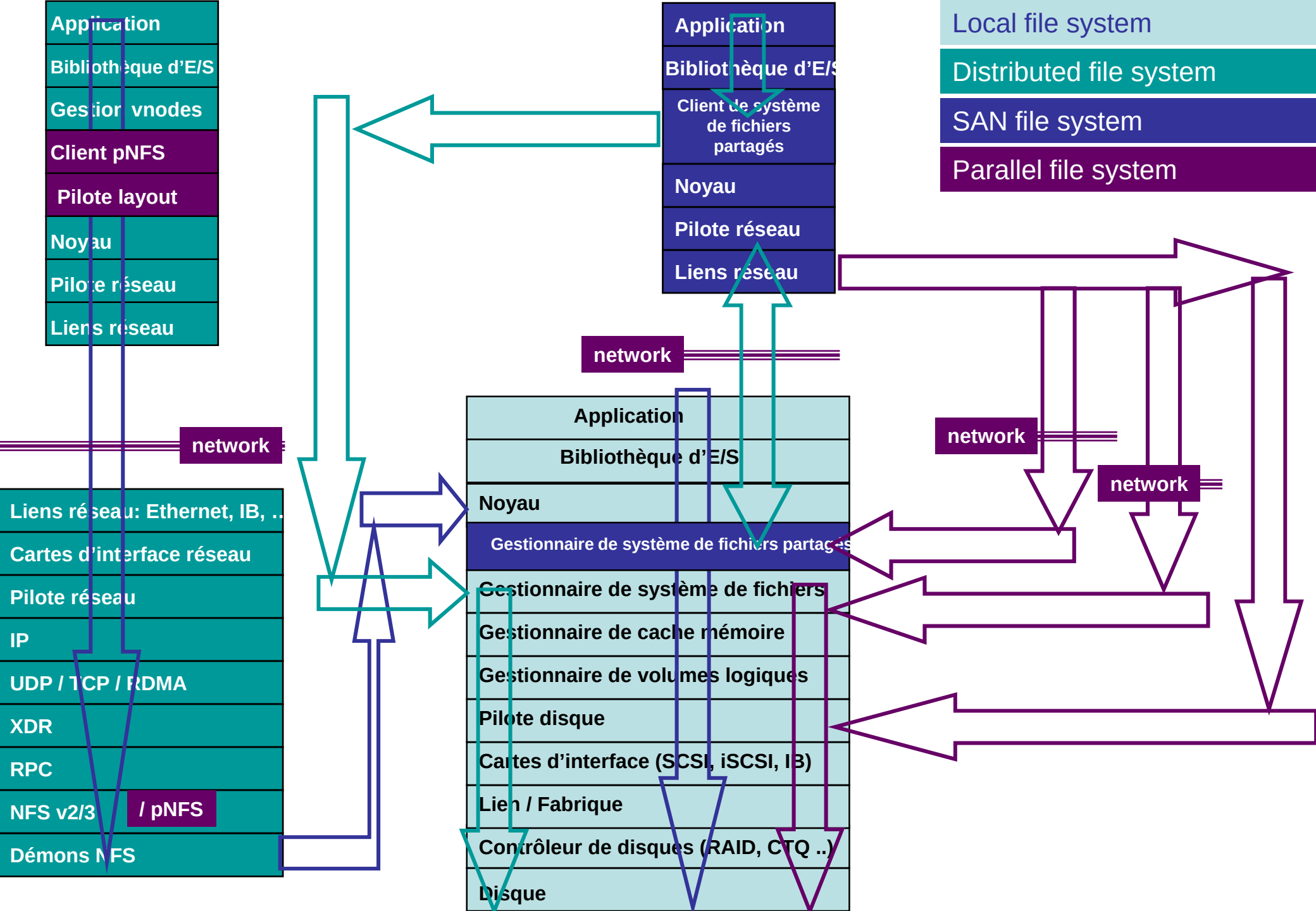
pNFS: Layouts

- One common client for different storage back ends
  - client retrieves a layout from the metadata server
  - layout maps file to storage devices and network addresses
  - client uses layout for I/O



Performance scalability limit (cont. 3)

- **Parallel file system**
- Metadata parallelism
  - **Solution** : possible, product architecture is not an obstacle
- Cache consistency improvement
  - **Solution** : nothing new in pNFS until now
- Higher complexity pNFS
  - Layouts management
  - **Solution**: Lustre, Panasas (one kind of layout)
- Protocol stacking
  - **Solution**: nothing, this is a cost for scalability; besides
    - Faster networks and NIC's
    - Faster disks and servers


Application
Bibliothèque d'E/S
Gestion vnodes
Client pNFS
Pilote layout
Noyau
Pilote réseau
Liens réseau
Application
Bibliothèque d'E/S
Client de système de fichiers partagés
Noyau
Pilote réseau
Liens réseau
Local file system
Distributed file system
SAN file system
Parallel file system
network
network
network
network
Application
Bibliothèque d'E/S
Noyau
Gestionnaire de système de fichiers partagés
Gestionnaire de système de fichiers
Gestionnaire de cache mémoire
Gestionnaire de volumes logiques
Pilote disque
Cartes d'interface (SCSI, iSCSI, IB)
Lien / Fabrique
Contrôleur de disques (RAID, CTQ ..)
Disque
Liens réseau: Ethernet, IB, ..
Cartes d'interface réseau
Pilote réseau
IP
UDP / TCP / RDMA
XDR
RPC
NFS v2/3
/ pNFS
Démons NFS

Agenda

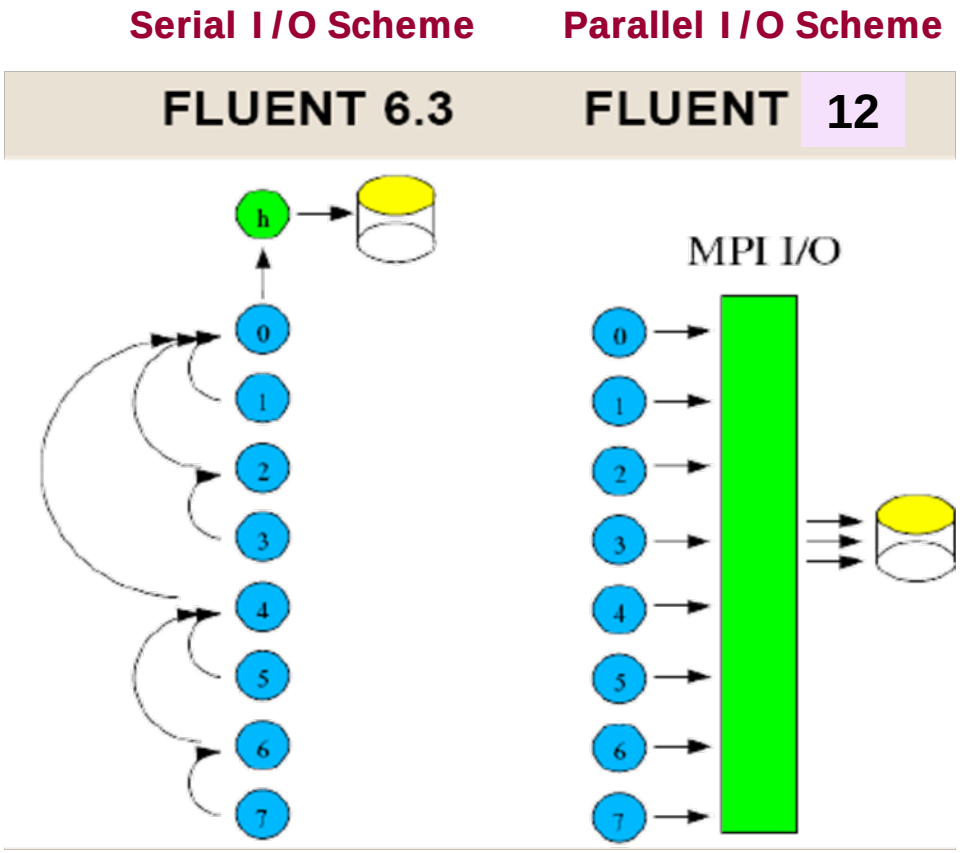
- Data sharing Evolution & current tendencies
- Performances: obstacles
- **Performances: some results and good practices**

Some performances on SGI systems

- Enhanced version of NFSv3
  - Ethernet based: Scaling to 16 CPU, 16 GigEthernet ports, and 1.8 GB/sec
  - IPoIB and NFS RDMA
    - read: up to 3.7 GB/s
    - Write: up to 1.4 GB/s
- SAN file system
  - Demonstrated to >45GB/sec
- Parallel file system
  - Panasas: Scales to > 10 GB/second
  - Lustre: Scales to > 50 GB/second

Parallel I/O Requires Parallel Storage: example FLUENT 12

Panasas and ANSYS Alliance Has Produced Parallel I/O for FLUENT 12



FLUENT 12: Offers support for Panasas

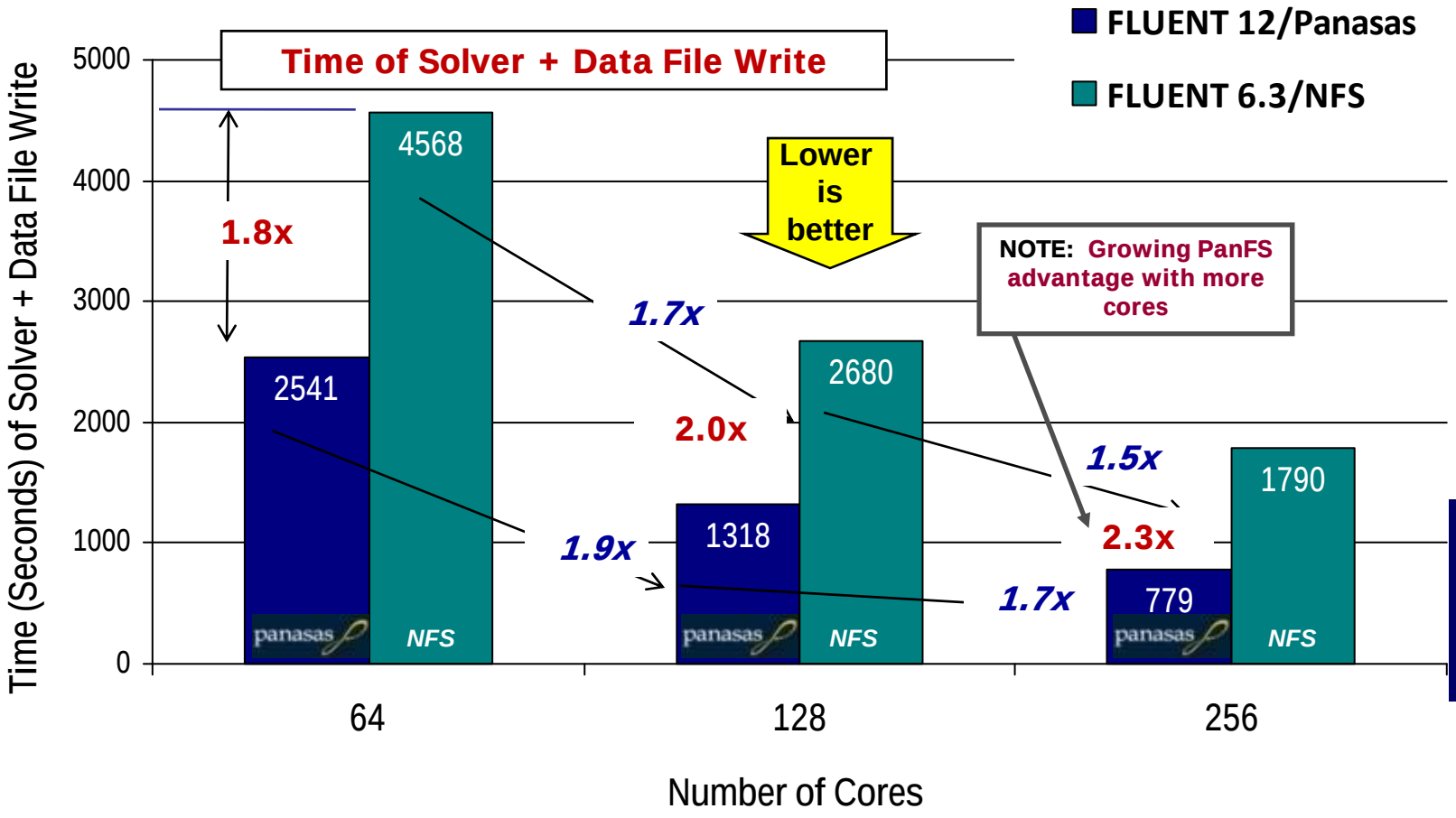
Commercial Release in April '09

Source: Barb Hutchings Presentation at SC07, Nov 2007, Reno, NV

Single Job: Scalability of Solver + Data File Write

FLUENT Comparison of Panasas vs. NFS on University of Cambridge Cluster

Unsteady external aero for 111 MM cell truck; 5 time steps with 100 iterations, and a single *.dat* file write



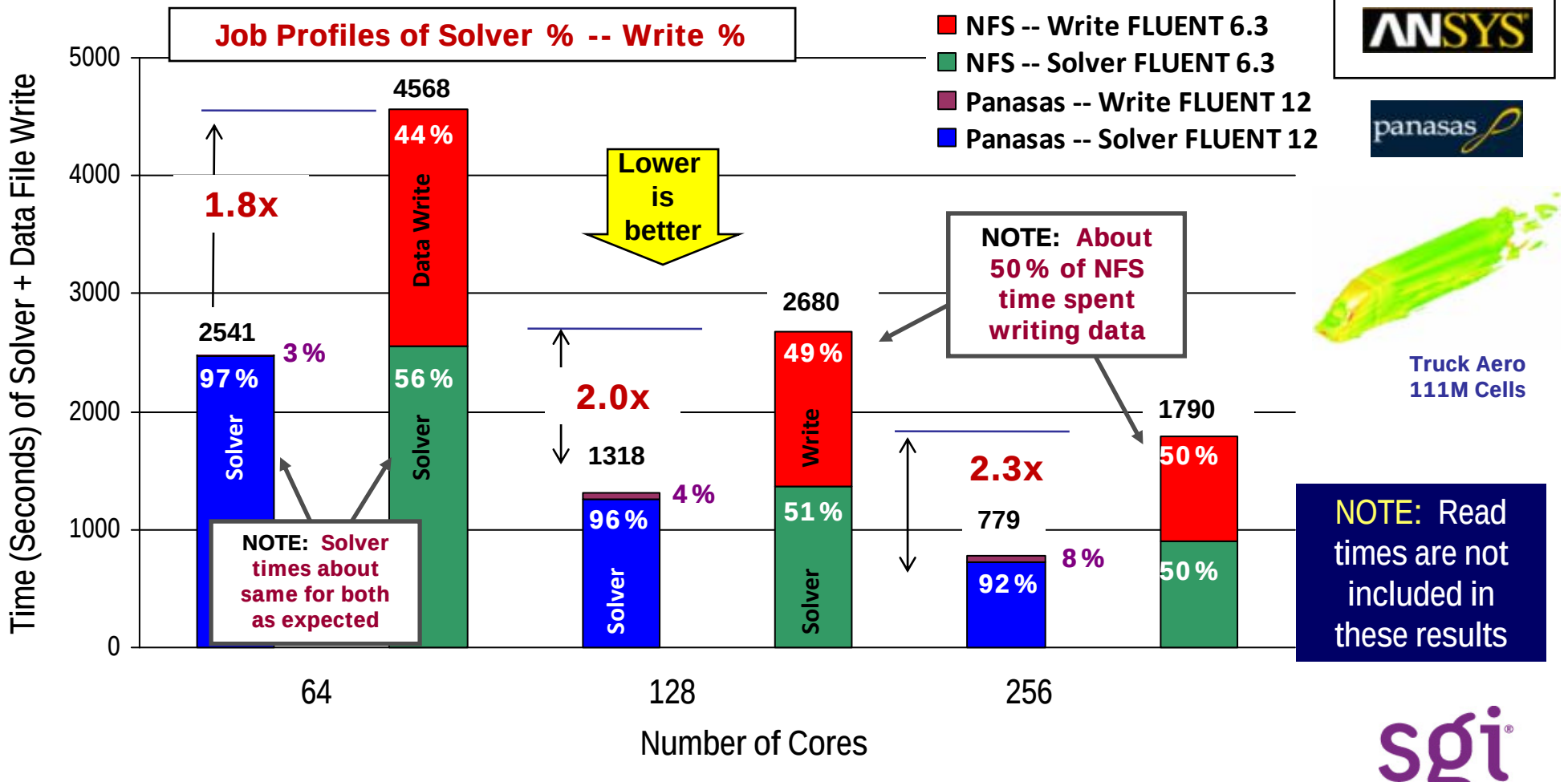
Truck Aero
111M Cells

NOTE: Read times are not included in these results

Single Job: Computational Profile of Solver vs. Write

FLUENT Comparison of Panasas vs. NFS on University of Cambridge Cluster

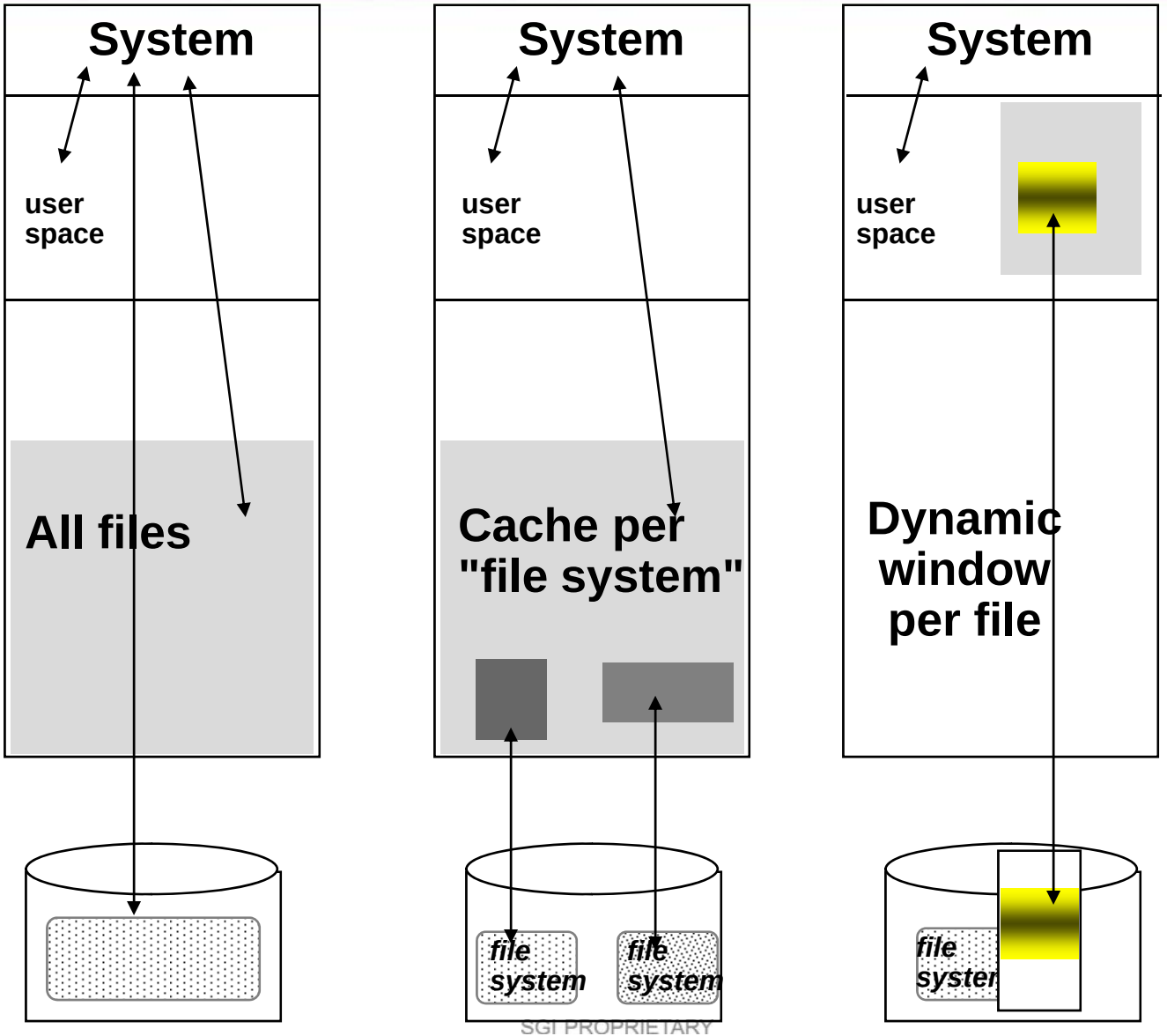
Unsteady external aero for 111 MM cell truck; 5 time steps with 100 iterations, and a single *.dat* file write



Whenever the source code cannot be changed

- Tuning of the I/O infrastructure (network included)
- Tuning at library level
  - MPI-IO
  - Application level memory Caches (the most efficient): FFIO at SGI
  - Better use proprietary client of the Parallel file system rather than a standard, poorly implemented
    - Lustre
    - DirectFlow (Panasas)
    - No NFS!
    - Lower latency
    - Higher scalability

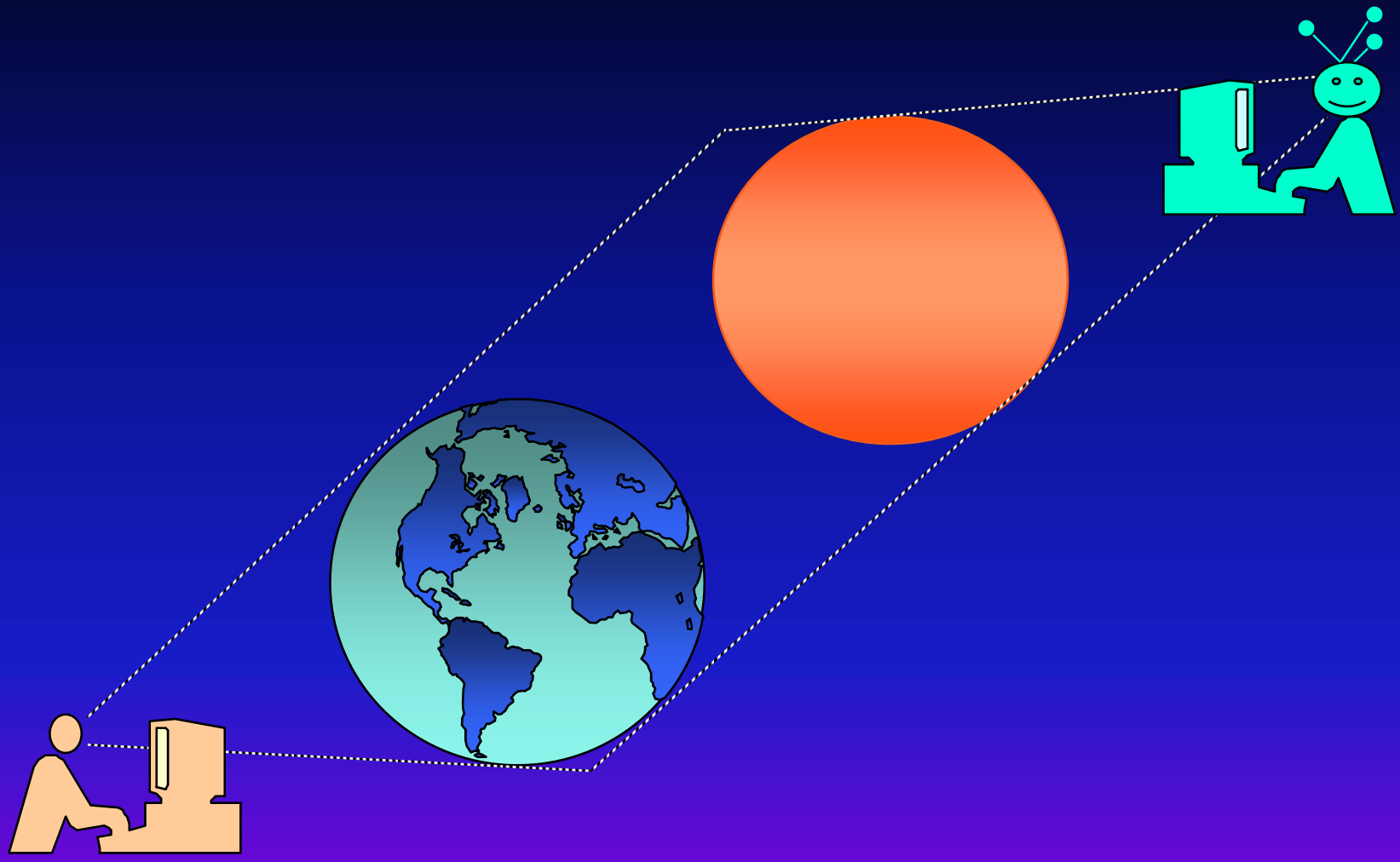
Whenever the source code cannot be changed : caches



Whenever the source code can be changed

- **Parallelize** I/O operations
  - Can be directed to all physical storage (OSS inside Lustre, StorageBlades inside Panasas)
  - Simultaneously
  - Independently
- File and i/o sizes
  - Higher is better (8 MB is optimal)
  - One i/o on a parallel file system ~= 100 ms (without cache)
    - Whether the i/o is 4 kB or 8 MB
- Do not share!
- Number of clients per storage entity
  - Best ratio optimal is one client per entity (StorageBlade in Panasas, in Lustre, a disk, a RAID group …)
  - Too many clients for one entity overload the entity
  - Too many entities for one client is not better
    - Parallel file system = network configuration
- Aim aggregate performance, do not bother with performance per client

Prospective future



Thank you for your attention