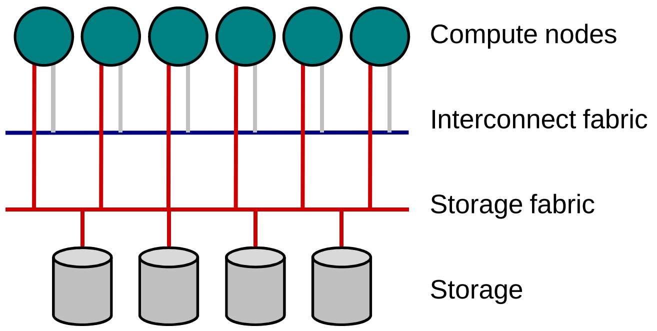
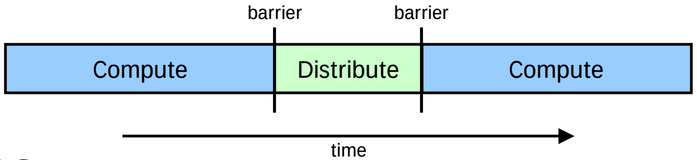
Extreme File Systems

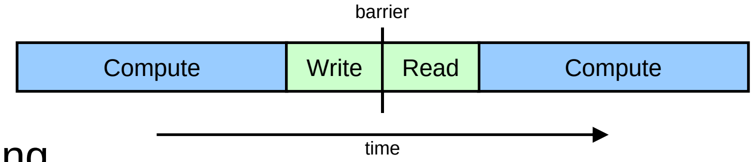
Roger Haskin
Senior Manager, File Systems
IBM Almaden Research Center

Parallel Computing 101 – File Systems for HPC

- **Application domain: large-scale simulation, climate modeling, weather prediction, petroleum, seismic, pharmacology,astrophysics, …**
- **Supercomputer is normally a large cluster whose nodes communicate over a high-speed fabric and share storage**
  - Nodes cooperate using shared memory, message passing, or shared storage to perform the computation
  - Computation often in phases: compute -> communication and I/O -> compute …
- **Any time spent doing file I/O is time wasted (i.e. time *not* spent computing)**
  - So file system performance is paramount.
- **Parallel file systems have become expected as the means to share storage within a computation and across workflows**
  - Single-system image simplifies programming
  - Posix semantics hides the complexities of clustering
  - Access modes:
    - Normally piecewise sequential
    - file per process and/or …
    - file per job (fine-grained read/write sharing within a file)



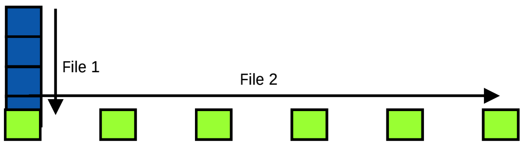


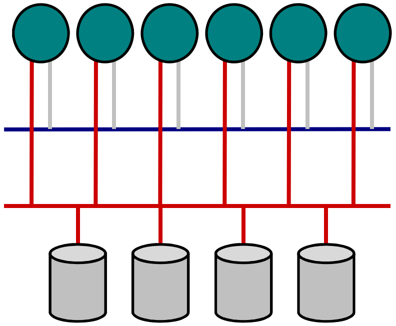


GPFS Concepts

- Shared Disks
  - All data and metadata on globally accessible block storage
- Wide Striping
  - All data and metadata striped across all disks
  - Files striped block by block across all disks
  - … for load balancing and throughput
- Distributed Metadata
  - No metadata node – file system nodes manipulate metadata directly
  - Distributed locking coordinates disk access from multiple nodes
  - Metadata updates journaled to shared disk

Principle: scalability through parallelism and autonomy

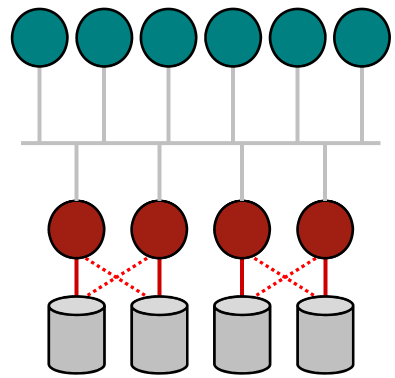




GPFS file system nodes

Control IP network

Disk FC network



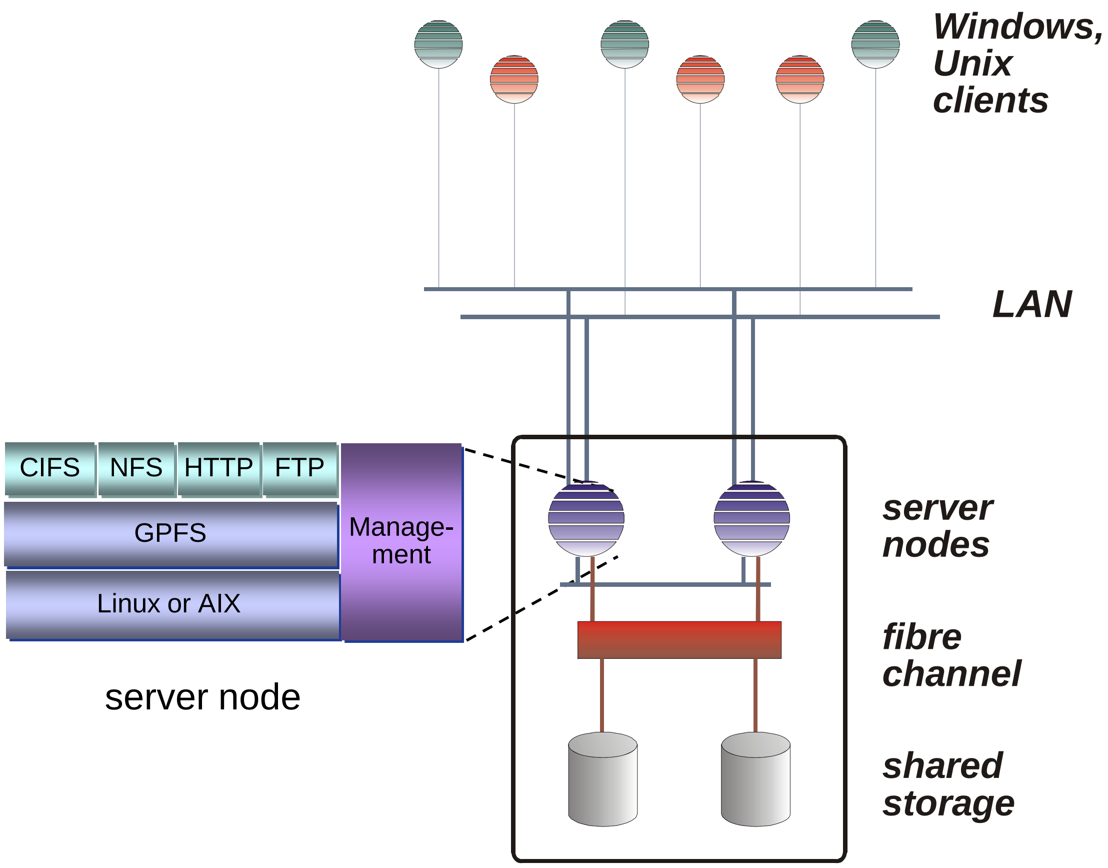
GPFS file system nodes

Data / control IP network

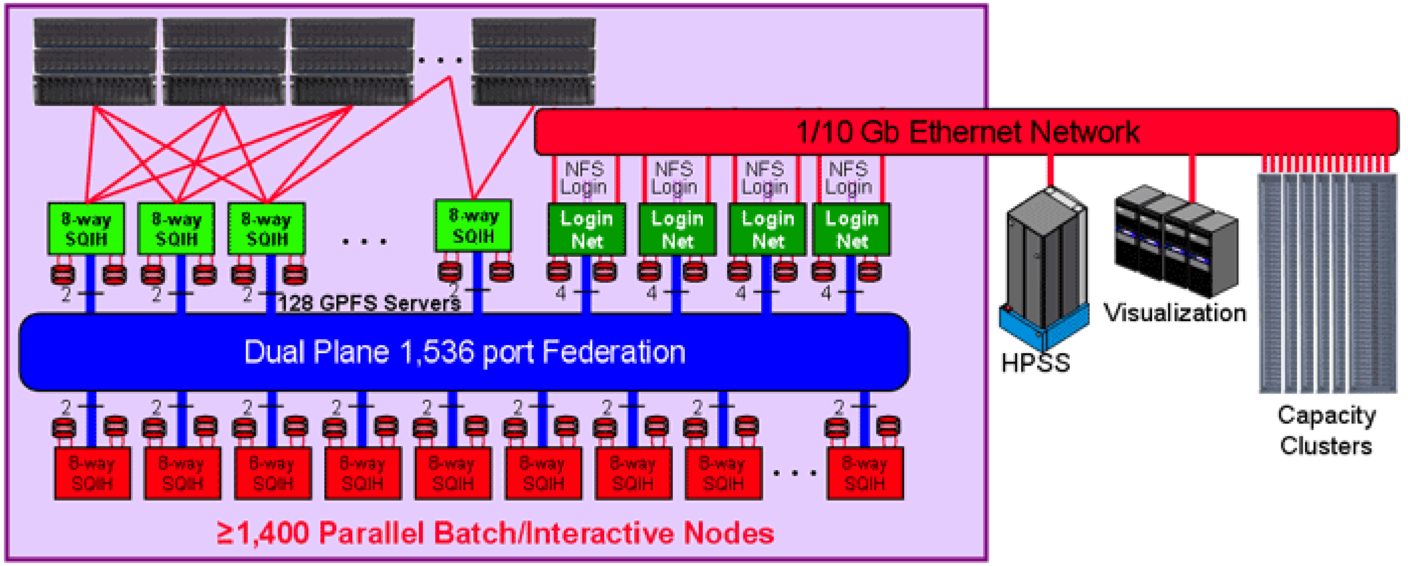
GPFS disk server nodes:
VSD on AIX, NSD on Linux – RPC interface to raw block devices

GPFS-based scale-out file server

- Problem: scalability limits of a conventional NFS or CIFS file server
  - Scaling by partitioning data across multiple, independent servers
  - Load and capacity balancing create a management nightmare
- Scale-out file serving with GPFS
  - Multiple server nodes share the same file system
  - Capacity and load balancing is automatic
  - Nodes and storage can be added incrementally
- Scale-Out File Server (SOFS)
  - Packaged version of Linux NFS/Samba/GPFS file server for scalable, data-intensive applications
  - IBM services offering, moving to product as marketing and support ecosystem matures
  - http://www-935.ibm.com/services/us/its/html/sofs-landing.html



GPFS on ASC Purple/C Supercomputer



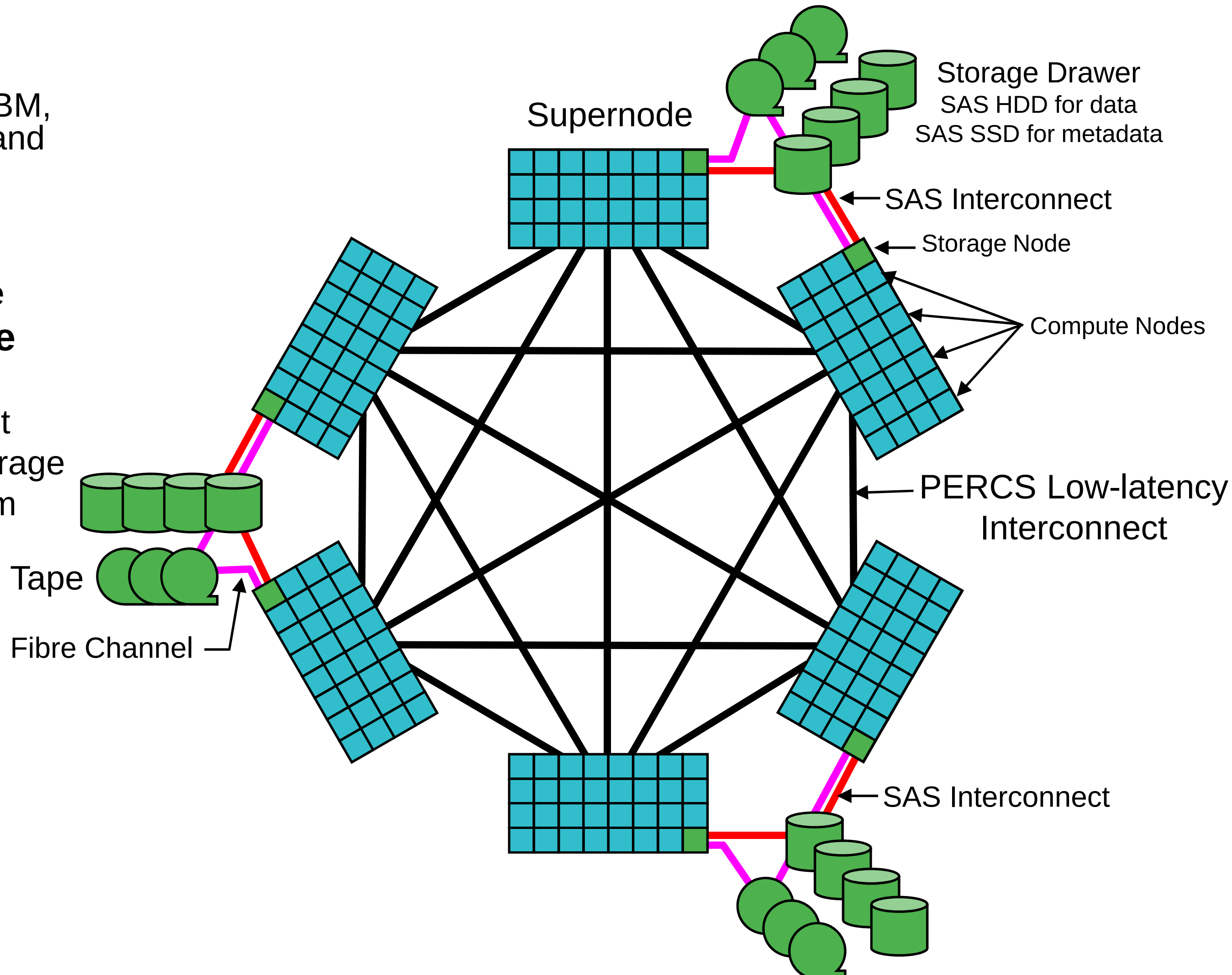
Purple System
- At least 1,400 parallel batch/interactive nodes
- 4 Login/network nodes from 2 SQH
- Clustered I/O with 128 SQIH for global I/O
- Dual plane 1,536 port Federation switch
- External networking
  - Login/network nodes for login/NFS/PFTP
  - All external networking is 1-10Gb/s Ethernet

Programming/Usage Model
- Application launch over all compute nodes
- 1 MPI task/CPU and Shared Memory, full 64b support
- Scalable MPI (MPI_allreduce, buffer space) to 8,192 tasks
- Likely usage
  - multiple MPI tasks/node with 2-4 OpenMP/MPI task
- Single STDIO interface
- Parallel I/O to single file, multiple serial I/O (1 file/MPI task)

- **1536-node, 100 TF pSeries cluster at Lawrence Livermore National Laboratory**
- **2 PB GPFS file system (one mount point)**
- **500 RAID conroller pairs, 11000 disk drives**
- **126 GB/s parallel I/O measured to a single file (134GB/s to multiple files)**

Blue Waters Supercomputer at NCSA

- **Blue Waters System**
  - NSF Track 1 program
  - Collaboration between IBM, NCSA, State of Illinois, and partners
  - Sustained petaflop
  - 200K processor cores
  - 10 petabytes file storage
- **IBM PERCS architecture**
  - Power7 processor
  - Low-latency interconnect
  - Shared memory and storage
  - GPFS parallel file system



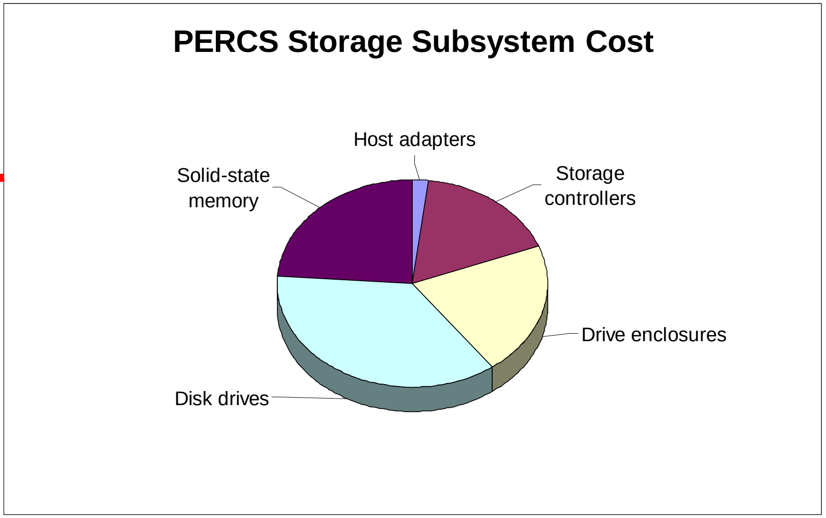
GPFS and PERCS

- **HPCS file system requirements (a subset)**
  - "Balanced" capacity and performance
    - (100 PB file system, 6 TB/s file I/O)
  - Reliability in the presence of localized failures
  - Support for full-up PERCS system (~64K nodes)
  - One trillion files to a single file system
  - 32K file creates per second
  - Streaming I/O at 30GB/s full duplex (for data capture)

5 years of iTunes music in 32 min!

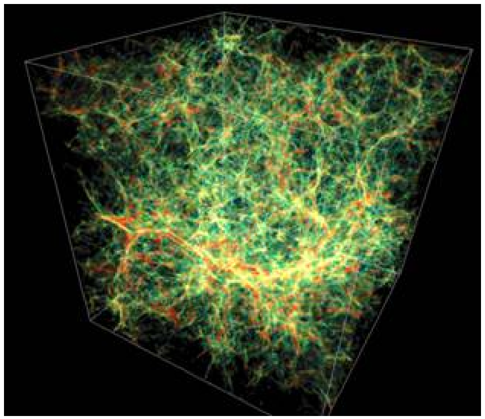
1PB of metadata!

- **Storage Requirements**
  - Reasonable cost - 10-20% of system cost
    - Large number of disk drives makes this difficult to achieve
    - Metadata performance requires substantial amount of expensive NVRAM or SSD
  - Reliability - system must continue to be available in spite of component failures
    - One or more drives continually in rebuild
    - Hard error rate between 10^{-14} and 10^{-16}
    - "Silent" data corruption
  - Productivity - non-disruptive repair and rebuild
    - Goal: rebuild overhead in the 2-3% range
    - Standard RAID rebuild can affect performance 30%
    - Parallel file system with wide striping: x% hit on one LUN causes same x% hit to entire file system

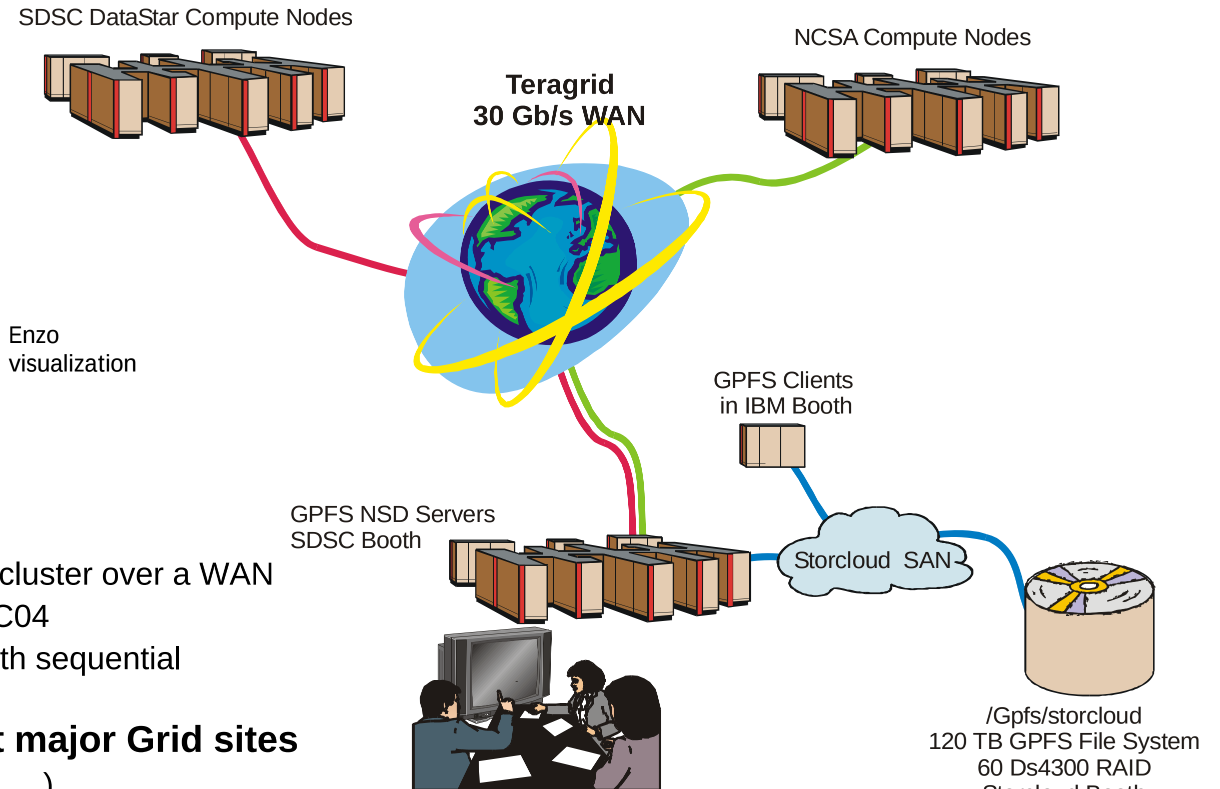


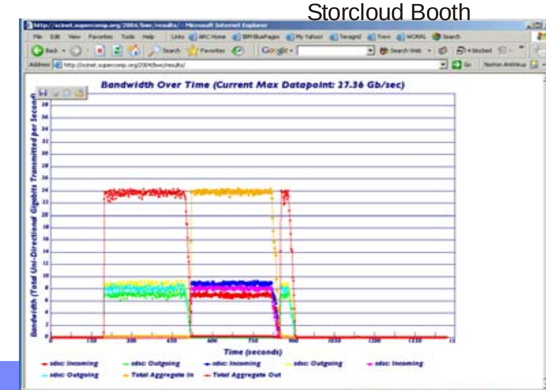
MTTDL 2 mo for RAID-5, 56 yr for RAID-6

GPFS over High-Speed WAN



Enzo visualization

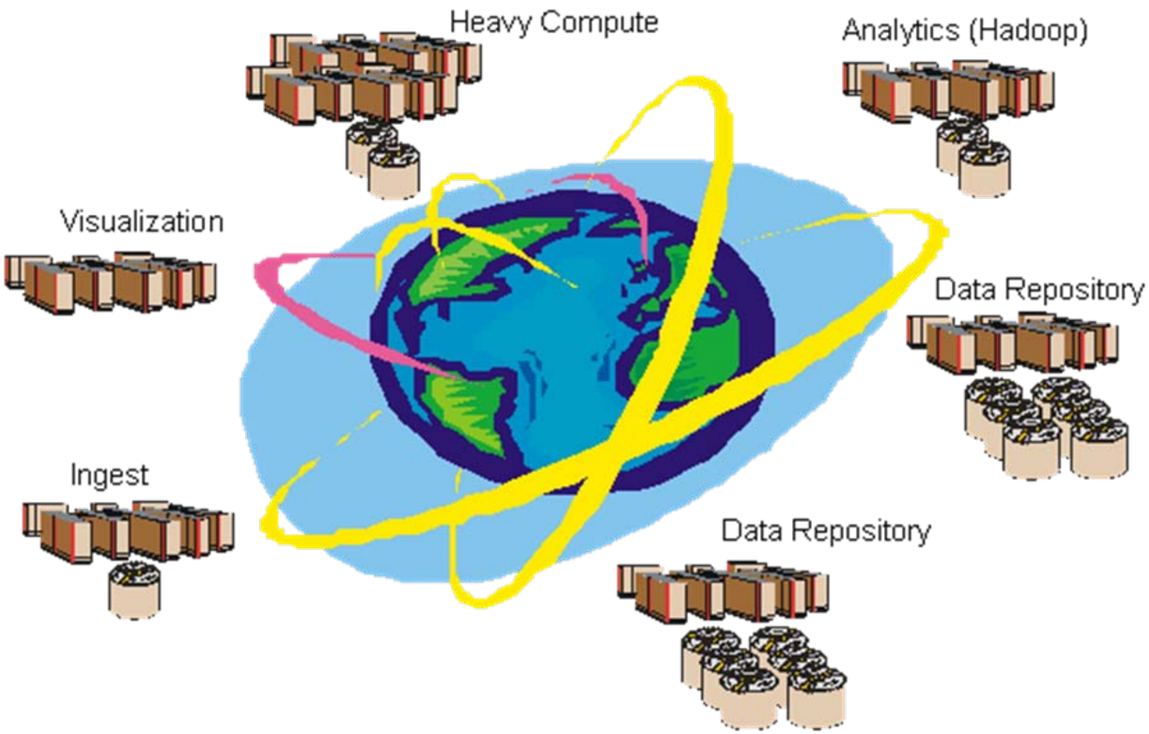




- **Multi-cluster GPFS**
  - Client cluster mounts from server cluster over a WAN
  - Technology demo at SC03 and SC04
  - Capable of saturating the WAN with sequential workloads and proper tuning
- **GPFS WAN in production use at major Grid sites**
  - Teragrid (SDSC, NCSA, Argonne, …)
    - 1500 nodes at 4 sites
    - 500TB shared file system
    - 30 Gb/s WAN backbone
  - DEISA
    - File systems at 8 sites
    - 10 Gb/s WAN backbone
- **1.5 GB/s continuous throughput achieved on Teragrid over many hours for real applications**

Panache: File system for the cloud

- **Cloud Storage**
  - Data is stored transparently in the cloud according to policies and access patterns
  - Endpoints are *SoFS servers* or *GPFS clusters*
  - *Panache* manages data placement and movement (in parallel) in the cloud
- **Why?**
  - Data can be efficiently accessed from anywhere
  - Sites can contribute resources to the cloud
  - Sites can be functionally specific (ingest, analyze, compute, repository, visualization, etc.)
  - Work can move around the cloud as needed - its data moves with it automatically
  - Allows a single copy of data, if desired, *but at the right place*
  - Allows data to be permanently replicated for performance, availability, fault-tolerance, disaster recovery
- **How?**
  - Each site has a local file storage (GPFS or SoFS), which serves as an entry point to the cloud and as a cache for remote data
  - Panache global cache directory tracks locations of all managed objects
  - Policies control placement and migration, e.g.
    - Data fetched from repository to endpoint on demand
    - Ingested data moved to repository, replicated for DR
    - Repository data archived or purged automatically



Extreme Fabrics for Extreme File Systems

What's important?

- **Performance**
  - Goes without saying!
- **Scalability**
  - Traditional SAN does not scale – switch bottlenecks, controller design
  - All large file systems use I/O nodes connected to compute fabric
- **Robustness**
  - GPFS has killed every fabric the first time it was used
  - Typically, fabrics are not designed for sustained high throughput
- **Standards, support for heterogeneity**
  - GPFS uses TCP for control traffic, supports OFED verbs (on Linux) and IBM proprietary fabrics on AIX.
    - Multiple fabrics can connect to storage through separate I/O nodes
  - Lustre has developed native support for multiple fabrics, basically implementing its own storage router. Difficult!

Questions?