High Performance Storage System Scalability: Architecture, Implementation and Experience

Dick Watson

Lawrence Livermore National Laboratory

925-422-9216

dwatson@llnl.gov

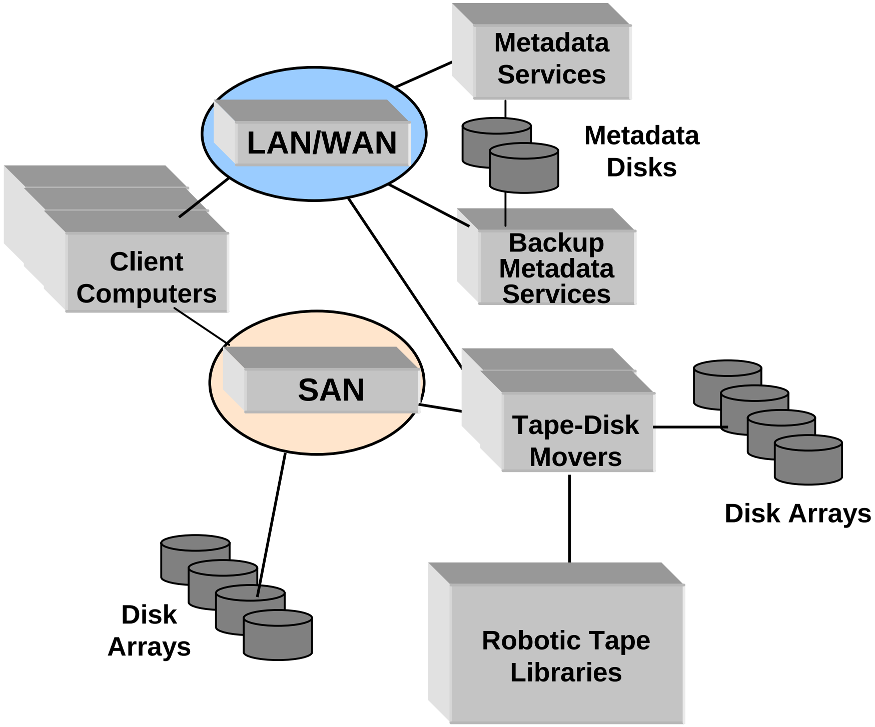
Development Partners

- Lawrence Livermore National Lab.
- Los Alamos National Laboratory
- National Energy Research Supercomputer Center
- Oak Ridge National Laboratory
- Sandia National Laboratories
- IBM Global Services in Houston TX

HPSS Web Site URL: **www.hpss-collaboration.org**

HPSS environment in a nutshell

- Hierarchical global file/HSM/archival system
- Distributed, cluster, network-centric architecture provides horizontal growth
- Devices SAN and/or LAN/WAN connected
- Metadata engine is IBM DB2
- Multiple storage classes
- Striped disks and tapes for higher data rates
- Multi-petabyte capability in a single name space
- Supports IBM AIX, Linux, Sun Solaris, and some SGI Irix components, mix and match



Scalability is crucial: yesterday, today and tomorrow

Parameter	Yesterday (1992)	Today (2005)	Tomorrow (2015)
Computing Power as Driver	10’s Gigaops	10’s - 100’s Teraops	10’s Petaops
Storage Capacity	10’s Terabytes	Petabytes	100’s Petabytes - Exabytes
Instantaneous Throughput	Megabytes/s	Gigabytes/s	100’s Gigabytes/s - Terabytes/s
Daily throughput	Gigabytes/day	10’s Terabytes/day	Petabytes/day

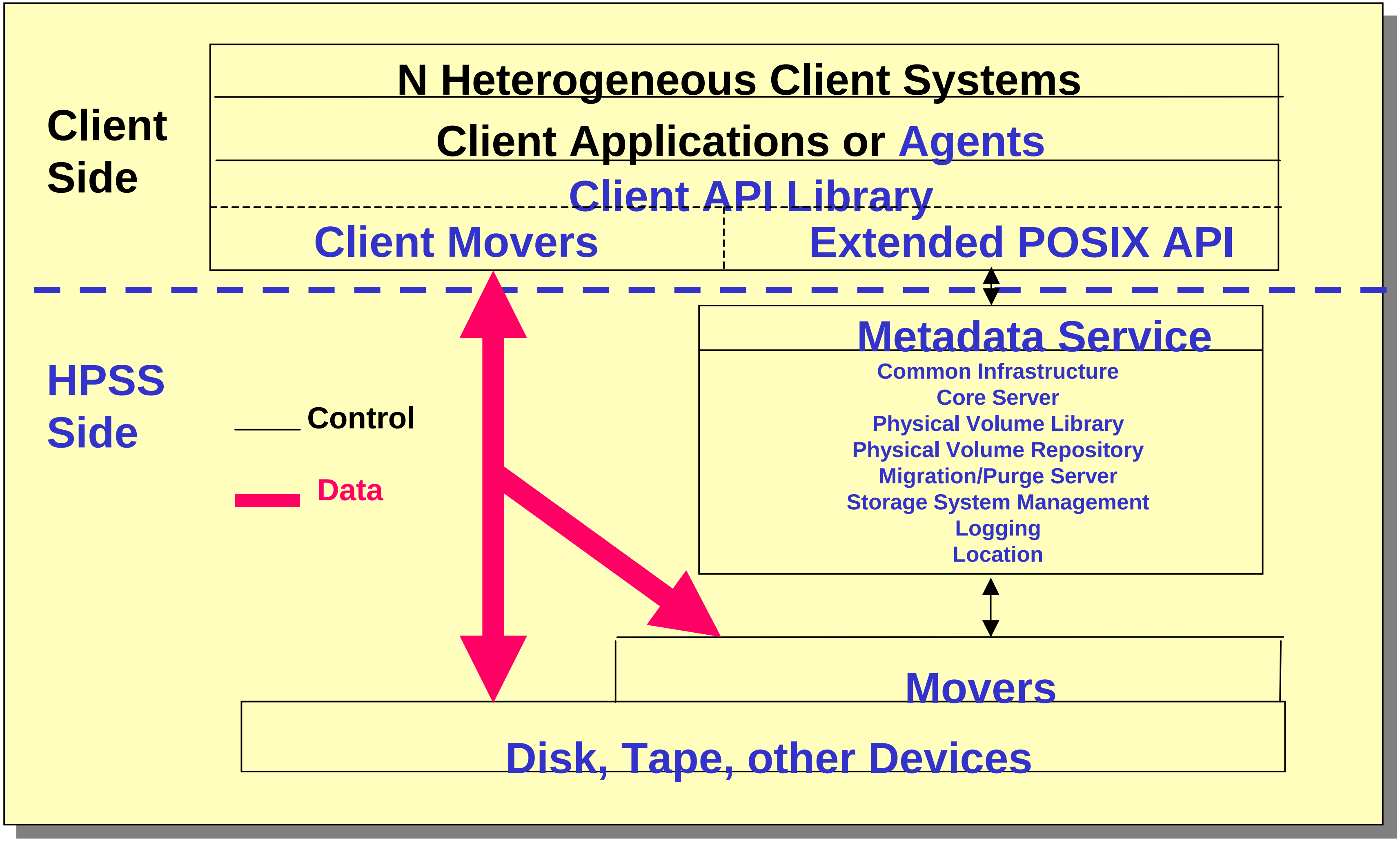
Three factors supporting scalability

- **Hardware**
  - Computational power
  - Networking
  - Storage capacity and I/O rate of media and controllers
- **Software**
  - Architecture
  - Implementation
- **Deployment**
  - Full attention end -to-end process
    - **Balanced configuration**
    - **Tuning**
    - **Planning**
    - **Support**

Scalability dimensions

- Scalable data I/O rate and throughput
- Scalable storage capacity and storage space management
- Scalable robustness
- Scalable name service
- Scalable numbers of clients
- Scalable deployment across geographical distances and multiple cooperating institutions
- Scalable storage system management
- Scalable security
- Client roles in scalability

HPSS high-level architecture:
(network-centric, robust metadata service)



HPSS second level architecture and implementation

- **HPSS Infrastructure**
  - Metadata Service
    - Scalable data structures and algorithms
  - Concurrency
  - Security Services
- **Communication Services**
- **Device Striping**
- **Storage Hierarchies, Classes-of-service, File Families**
- **Subsystems**
- **Client Interfaces**
- **No Kernel Modifications**

Metadata service

- **Advantages of asymmetric metadata architecture; it simplifies:**
  - Lock management for concurrent accesses
  - Metadata integrity, consistency, recovery
  - Security - all metadata is outside client systems, accessible through single authenticated message interface (minimizes components to be trusted)
  - Supporting HSM/archive services for many heterogeneous file systems
- **Issues with asymmetric metadata architecture:**
  - Single point of failure does require redundant computers
  - Some extra operation latency
- **Highly robust because:**
  - Widely used commercial RDBMS metadata engine
  - Atomic transactions
  - Metadata stored in separate mirrored RAID storage
    - Separating metadata from user data also supports fast metadata restore time independent of amount of user data
  - RAID disks backed up at least daily
  - Redundant metadata machine(s) with manual or automatic failover

Scalable data structures and algorithms

- **Particularly important to scalable capacity (storage space and number and sizes of objects)**
  - 64 bit field sizes and arithmetic
- **Dynamic allocation of metadata space and data structures**
- **Space allocated in large contiguous segments of variable length**
- **Free space management**
  - HPSS uses in memory data structures rebuildable from stable disk structures
- **Use of DB2 B+ trees, dynamic structures and space management**

Concurrency

- **Critical to support multiple concurrent operations throughout the system**
- **Uses IEEE POSIX threads**
- **Implemented with minimum units of metadata unit locking granularity**
- **All components implemented as multithreaded and thread safe:**
  - Enables HPSS to scale using clusters of multiprocessors

Communication services

- **SAN, LAN, WAN communication central to HPSS scalability**
  - Key is to support optimum use of each site's networking infrastructure
- **Multiple modular levels of communication capabilities in HPSS**
  - Basic Networking - currently TCP/IP GigE and 10GigE dominant (earlier support included HIPPI) - also SAN support (FC and iSCSI) and support for cluster internal networks (e.g. Quadrics, Myrinet)
    - Tuning networking (e.g.TCP/IP) parameters such as packet and buffer sizes crucial
    - Multiple networking technologies and configurations at a site
  - Control communication uses RPC, while Mover-Mover communications uses socket communication
  - Mover-to-Mover data transfer protocol supports negotiated transfer optimizations
  - Network striping using multiple TCP connections, multiple NICs, multiple cluster nodes (Client and HPSS sides), Client agents have access to configuration information and have algorithms to optimize use of the above

Device striping, storage hierarchies, classes-of-service, file families

- **Virtual volume service includes striping files on HPSS supported disks and tapes to individual files**
- **HPSS supports the ability to organize classes of devices into multiple storage hierarchies and device stripe widths**
  - Central to scaling of capacity and I/O - Allows different classes of devices to be organized into different classes-of-service (COS) for cost or performance
  - Example at LLNL 5 COS used, each using a different hierarchy of devices: Small files (<4MB), Medium files (4MB - 32MB),Large files (32MB - 256MB), Jumbo/htar (>256MB), dual-critical (large/jumbo files that are mirrored to tape)
- **File families assure files in a directory subtree can be collocated on the same media volumes**
- **The above can be coupled with site dependent staging and migration/purge policies**
  - Files that are written are marked thus eliminating need to search whole metadata structure for migration candidates

Multiple levels of user interfaces

- **Basic interface is an extended POSIX API (CLAPI) supporting parallel I/O, COS and other HPSS functions**
  - CLAPI can be used directly by client applications or data service applications (client agents)
- **VFS, Local File Mover, CIFS via SAMBA and NFSv4 support**
- **XDMS (DMAPI) interface is supported to XFS**
- **Transfer agents help achieve optimum I/O transfers. This can be a complex configuration dependent problem:**
  - Multiple networks, NICs, nodes, parameters, stripe widths, resource allocation, error recovery, transfer job restart, debugging etc.
  - Example client agents available with HPSS or have been written by sites with configuration knowledge for transfer optimization:
    - Parallel File Transfer Program (PFTP), Hierarchical Storage Interface (HSI) and Parallel Storage Interface (PSI)

I/O scaling using scalable-units and tuning

- **Scaling is a continuous process as new hardware and media are introduced and requirements change**
- **Successful scaling is an end-to-end problem!**
  - Many interacting components, parameters, protocols etc
    - E.g OS’s, file systems, HPSS segment size, and stride length, buffer and packet sizes, network topologies and technologies, even controller microcode and direction of transfer
  - Finding the sweet spot is non trivial
- **An optimal combination of Mover, NICs, set of devices, parameters, comsys56l slduce ngeS)25**

Scalable data throughput

- **Architecture**
  - Separation of data and control and use of Movers
  - Storage service and its virtual volume service (e.g. striping)
- **Implementation**
  - Concurrent requests and I/Os
  - Modular set of communication services including intelligent client agents
  - Device striping
- **Deployment**
  - Scalable-units
  - Use of commodity multiprocessor clusters
  - Periodic I/O planning
- **Issues needing work**
  - Improve small file performance (e.g. # of creates/s and read-writes/s)

Scalable capacity

- **Architecture**
  - Hierarchical storage architecture
  - Multiple hierarchies, COS and file families
  - Separation of migration/purge policies and mechanism
- **Implementation**
  - Metadata engine choice and scalable metadata design and organization
  - Scalable data structures
- **Deployment**
  - Periodic review of storage requirements and technologies
- **Issues needing improvement**
  - None identified

Capacity scaling examples

- **1.7 PB** Lawrence Livermore National Lab (LLNL) Secure Computing Facility (SCF) (**~28 million files**) **scaled from13 TB** in 1992.
  - **1.4 PB** LLNL Open Computing Facility (OCF) **(~20 million files)**.
  - **~1 million directories in the OCF and 0.5 million in the SCF (10K - 90K entries)**.
- **2.8PB**: Los Alamos National Laboratory (LANL) SCF, **(~ 38M files).**
- **2+PB**: Brookhaven National Laboratory (BNL).
- **1+PB**: Commissariat à l'Energie Atomique/Division des Applications Militaires (CEA/DAM) Compute Center in France.
- **2+PB**: The European Centre for Medium-Range Weather Forecasts (ECMWF) in England.
- **1.5PB**: National Center for Environmental Prediction (NCEP)
- **1+PB**: National Climate Data Center (NCDC)
- **1+PB**: National Energy Research Scientific Computing Center (NERSC), in **33M** files.
- **1.5PB**: San Diego Supercomputer Center (SDSC).
- **1.4PB**: Stanford Linear Accelerator Center (SLAC).
- Many sites, such as ORNL, are **doubling their stored data yearly** and will also shortly reach a **petabyte**.

I/O Scaling Examples

- **LLNL** - Aggregate data transfer rates to the archive, before HPSS, were well under **10MB/s** and now exceed **1.5GB/s** to caching disk. Single file rates, using a four-way stripe to a RAID array, generally run at around **300 MB/s**. Daily throughput to the archive has exceeded **17 TB/day**.

- **LANL** - A recent user archive operation stored **122,000 files occupying 10TB in six hours** with the transfer rate limited by network throughput. In a recent performance demonstration, a data transfer rate of **550 MB/s was achieved using 16-way mirrored tape stripes** storing files over 100 GB in size on StorageTek 9940Bs.

- **LBNL** - NERSC has gone from moving **1.5TB/day in 2001** to peak I/O days of **6TB/day in 2004**, with expected peak days of **10TB/day in 2005**. Single file transfer throughput has gone from **17MB/s in 2001 to 231MB/s in 2004**, limited by network bandwidth.

- **BNL** - Daily ingest rate from experimental devices to HPSS has reached **28TB/day**, and **330MB/s and 550MB/s I/O to tape and disk** respectively.

- **IBM** - At the SC04 supercomputing conference in November 2004, IBM demonstrated HPSS (an early version of HPSS 6.2) performance using three computers, one each for HPSS, reading and writing. A large 128 GB file was written and read in 512 MB blocks using **16-way striped SAN-attached disk files**, using 8 host bus adapters on each client computer. As one computer wrote each block, it was immediately read by a second computer, thus demonstrating **"read behind write"** performance. The **file transfers were measured at 1016 MB/s on the write side and 1008 MB/s** on the read side, for an aggregate data rate of just over two GB per second.

Conclusions

- **The modular network-centric, distributable architecture of HPSS and modular industry standard product infrastructure are sound.**
- **HPSS has demonstrated the following scaling factors:**
  - 100 for capacity to petabytes,
  - 1000 for instantaneous throughput to GB/s,
  - 1000 for daily throughput to 10s TB/day, and
  - 1000 for single file bandwidth to GB/s.
- **The HPSS system architecture and implementation has lots of room for further scaling in I/O, capacity and other dimensions by further orders of magnitude in the future.**
- **The future near term scaling focus will be on measurement, tuning and optimization, particularly metadata performance, thus improving small file performance and supporting other scalability dimensions.**