# Yes Virginia, There is an HPSS in Your Future

**Dick Watson**

***Lawrence Livermore National Laboratory***

**925-422-9216**

**dwatson@llnl.gov**

**Development Partners**

– Lawrence Livermore National Laboratory - Oak Ridge National Laboratory
– Los Alamos National Laboratory - Sandia National Laboratories
– National Energy Research Scientific Computing Center - IBM

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

*Prepared for Salishan Conference on High Speed Computing, Salishan Oregon, 4/24-27/2006*

HPSS

UCRL-PRES-220462

# Yes Virginia, There is an HPSS in Your Future

- **The HPSS Collaboration has a roadmap to meet Petascale environment HSM/Archive requirements with relatively few changes.**

- **To quote Tom Ruwart (I/O Performance, Inc.)**
  - "HPSS is a mature, scalable, and reliable *architecture* that can be easily adapted to meet petascale computing requirements
  - It is easily adapted to support object-based systems and devices. Object concepts are inherent in the HPSS architecture
  - It's use of an enterprise-class relational database will allow HPSS to gracefully and robustly scale beyond a trillion files and a very large global name space
  - Why start something new when HPSS has a 15 year head start?"

- **Talk outline**
  - Review why HSM/archives are economically required in HPC environments.
  - Review the HPSS architecture, capabilities and scalability experience.
  - Outline the HPSS roadmap to meet Petascale environment requirements.

HPSS

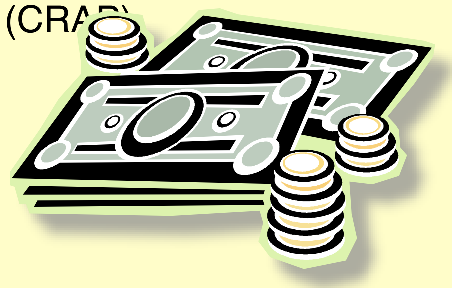
# Review: Why HSM/archives are economically required in HPC environments

# Need to look at the total cost

**One hears new "common wisdom":** *Disk is cheap, tape is dead*

***Total Cost of Ownership/Performance not*** **unit Price/Performance is the key**

- **Need to take into account all cost factors**
  - Purchase price (balanced reliability, capacity and I/O)
    Big interest as one wag notes in consumer reliability and price (CRAP)
  - Recurring maintenance
  - Power
  - Cooling
  - Administration/management
  - I/O Infrastructure to balance capacity and I/O requirements
  - Footprint
  - Site/specific and political…



- **Do your own study/analysis for your installation (I'd like to know what you find)**

# Example: tape is much less expensive than disk* - it really adds up for Petabytes -

HPSS

- **In the *LLNL, LANL* environments** tape is:**
  - *6.7X, 54X* cheaper to purchase (including drives, robotics, movers and media).
  - *56.7X, 14X [currently under warranty]* cheaper than disk for yearly maintenance
  - *72X, 105.5X* cheaper net yearly upkeep
  - *342X, 722X* cheaper than disk for electrical power to keep them spinning
  - *342X, 722X* cheaper for cooling (~1/3 total cost of power above)

Estimated total disk power cost is in the range $500K-700K/PB/yr

**Data obtained in 2005*
*** Differences primarily reflect different equipment*

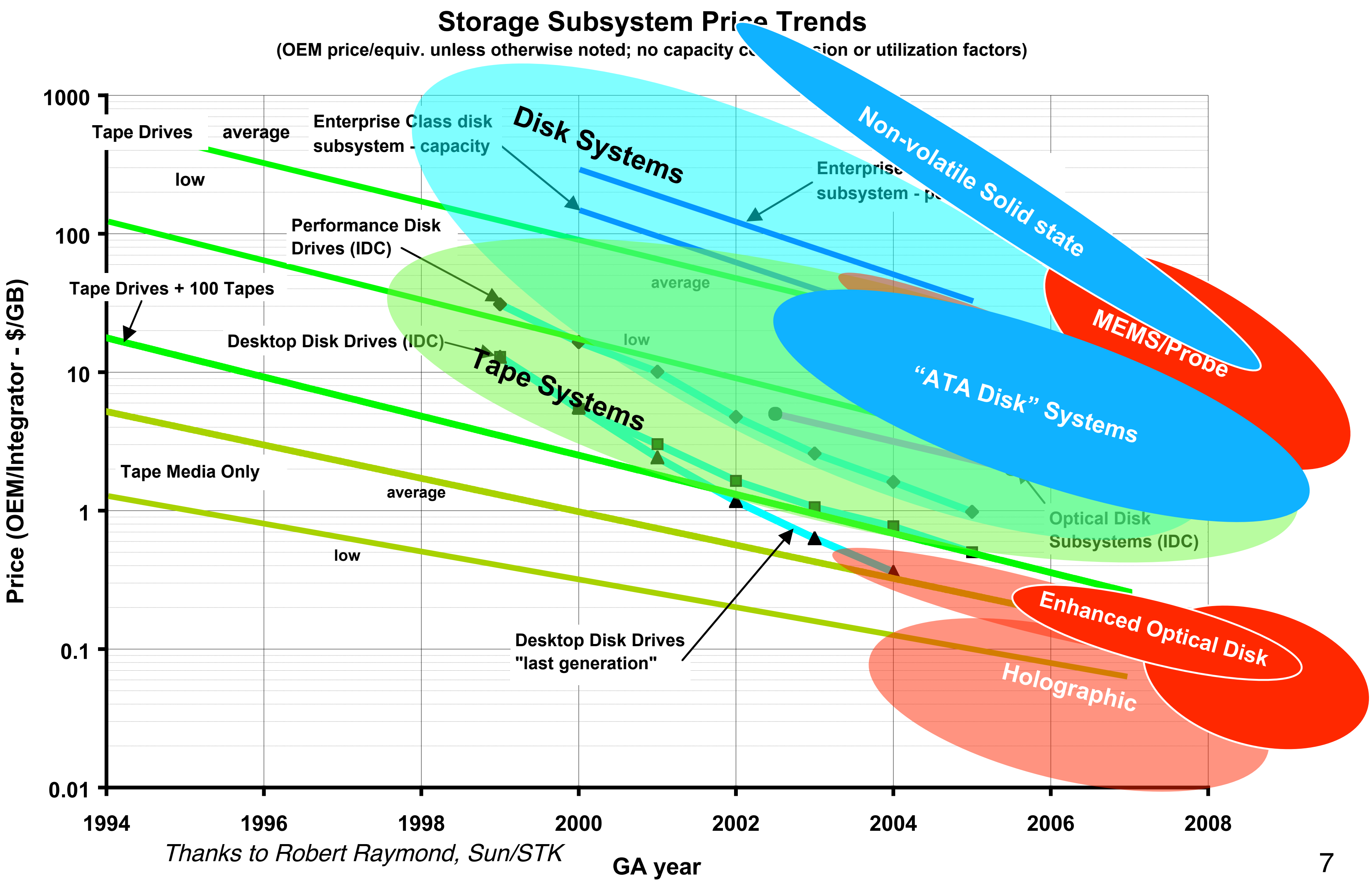
# Storage device futures: no significant surprises expected, most technologies on their evolutionary tracks

- **Magnetic disk recording density progress slowdown**
  - Rate of advancement of magnetic disk operating point demos slows to 27% CAGR (products 29%)
  - 14 sq in on 3.5 in disk, currently .1Tb/sq, @30% CAGR reach superparamagnetic limit 8 yrs. approaching top of S-curve
  - More disk drive product differentiation and specialization – in lieu of traditional density progress
  - MAID looking for application space in HPC. Relatively expensive, data lifetime questions.
  - Removable disk in tape cartridges. Data lifetime questions.
- **Tape is not dead and can maintain its cost/GB advantage over disk (e.g. NSIC tape roadmap shows linear growth to 2015 (16TB/cartridge and 833MB/s))**
  - 14,000 sq in on tape cartridge, currently .00044 Tb sq ", 41%/yr CAGR capacity growth, no limit in 8 yrs. On mid steep slope of S-curve
- **Consumer products are fueling significant solid state memory price erosion (<$30/GB) with miniature magnetic disk "threatened"**
- **Optical disk consolidating on blue laser, DVD derived technologies – "blu-ray" devices available – roadmaps to 200 GB/disk, relatively low data rates ~10MB/s**
- **No holographic based storage systems available this year but more progress made – prototypes demonstrated, "HVD" holographic versatile disk standard introduced (as a wag says "it's the future, always has been, always will be")**
  - Vendors could package in cartridges for use in existing robotic libraries
- **MRAM low capacities, lithographic limits.**
- **MEMS, molecular storage, other new storage technologies in early research, many years away, if ever**

*Thanks to Robert Raymond,Sun/STK: Gordon Hughes, UCSD: Dave Anderson, Seagate*

# Economics drives technology choices

HPSS




*Thanks to Robert Raymond, Sun/STK*

# Summary: Why need HSM/Archives

- ***Uses the most cost effective storage***
  - **Today tape is most affordable/sustainable media for large archives**
- **Machine/OS/file system agnostic storage solution**
- **Provides cost effective long-term data stewardship (ILM)**
  - Protection of billions of dollars of data investment
  - ***Outlive vendors, machines, operating systems, file systems***
  - Protection from platform disasters (software or hardware)
  - Repack and data rescue tools for information lifetime management
  - Multiple copies
- **Risk-averse solutions not tied to "latest" changes (e.g. OS releases, maintenance) on compute platforms**
- **Scales larger than most file systems - #files, directories, file sizes**
- **Intelligent resource usage/data placement**
  - Classes-Of-Service,
  - Stage/migrate/purge
- **Robotic/atomic mounts of sequential media**
- **Access to devices that have long inherent delays**
- **Any storage product (e.g. Object Storage Devices) can be used in HSM/archives where it makes economic sense.**

# Review: HPSS architecture, capabilities and scalability experience

More details in: Watson, R.W. 'High Performance Storage System Scalability: Architecture, Implementation and Experience," 22nd IEEE - 13th NASA Goddard (MSST2005) Conference on Mass Storage Systems and Technologies April 11-14, 2005, Monterey, California.

**http://storageconference.org/2005/presentations/watson.pdf**

# Scalability is crucial: yesterday, today and tomorrow

HPSS

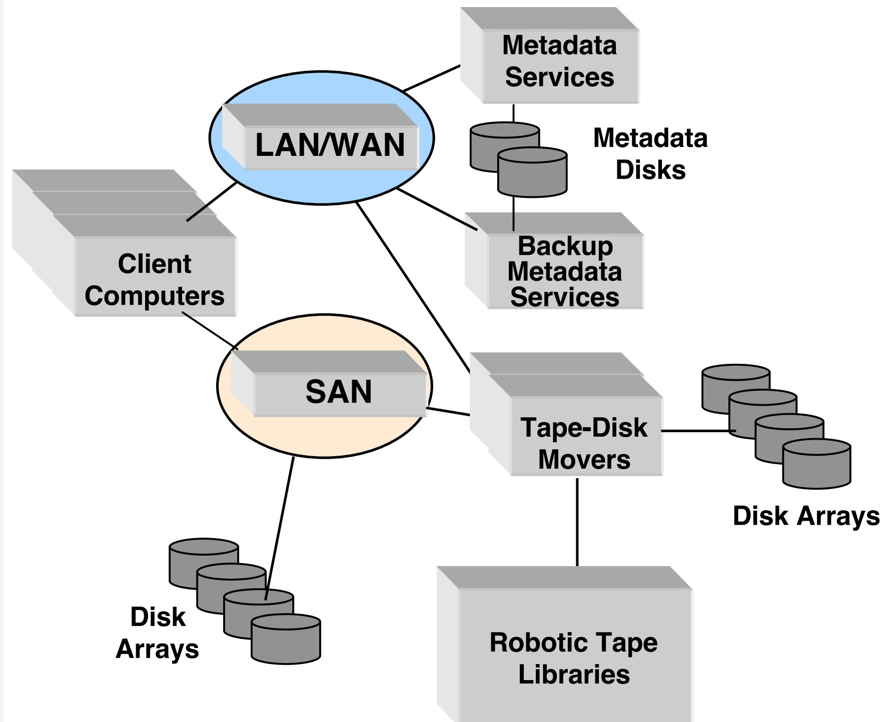
| Parameter | Yesterday (1992) | Today (2006) | Tomorrow (2015) |
|---|---|---|---|
| Computing Power as Driver | 10’s Gigaops | 10’s - 100’s Teraops | 10’s Petaops…. |
| Storage Capacity | 10’s Terabytes | Petabytes | 100’s Petabytes - Exabytes |
| Instantaneous Throughput | Megabytes/s | Gigabytes/s | 100’s Gigabytes/s - Terabytes/s |
| Daily throughput | Gigabytes/day | 10’s Terabytes/day | Petabytes/day…. |

# HPSS meets key scalability and other requirements

| | |
|---|---|
| **Reliability/recoverability/availability**<br>• Metadata-mirrored,backup,recovery<br>• Atomic transaction<br>• Multiple file copies | ✓ |
| **Performance (separation of control and data)**<br>• I/O rate to single file GBs/s and beyond<br>• Direct client to HPSS parallel I/O<br>• Simultaneous throughput GBs/s and beyond<br>• Transaction rate | ✓<br>✓ *(Today metadata performance scalability is by static subtrees)* |
| **Capacity**<br>• Global name space and data sharing (LAN/SAN/WAN access)<br>• 10s Petabytes and beyond (flexible expansion granularity)<br>• # files billions (unlimited file size)<br>• # directories billions and beyond (unlimited directory size)<br>• Automatic migration/staging<br>• Multiple, multilevel hierarchies, Classes of Service, file families | ✓ |
| **Transparency, user interfaces, file system integration**<br>• Access methods (e.g. Posix, PFTP, NFS, PSI, HSI, HTAR, VFS)<br>• File System integration (VFS, DMAPI, Lustre, Panasas, GPFS, other) | ✓ |
| **Security**<br>• Authentication, Posix permissions, ACLS<br>• Secure network transfers | ✓ |
| **Others (e.g. extendibility by labs, manageability, # of clients)** | ✓ |

# HPSS environment in a nutshell

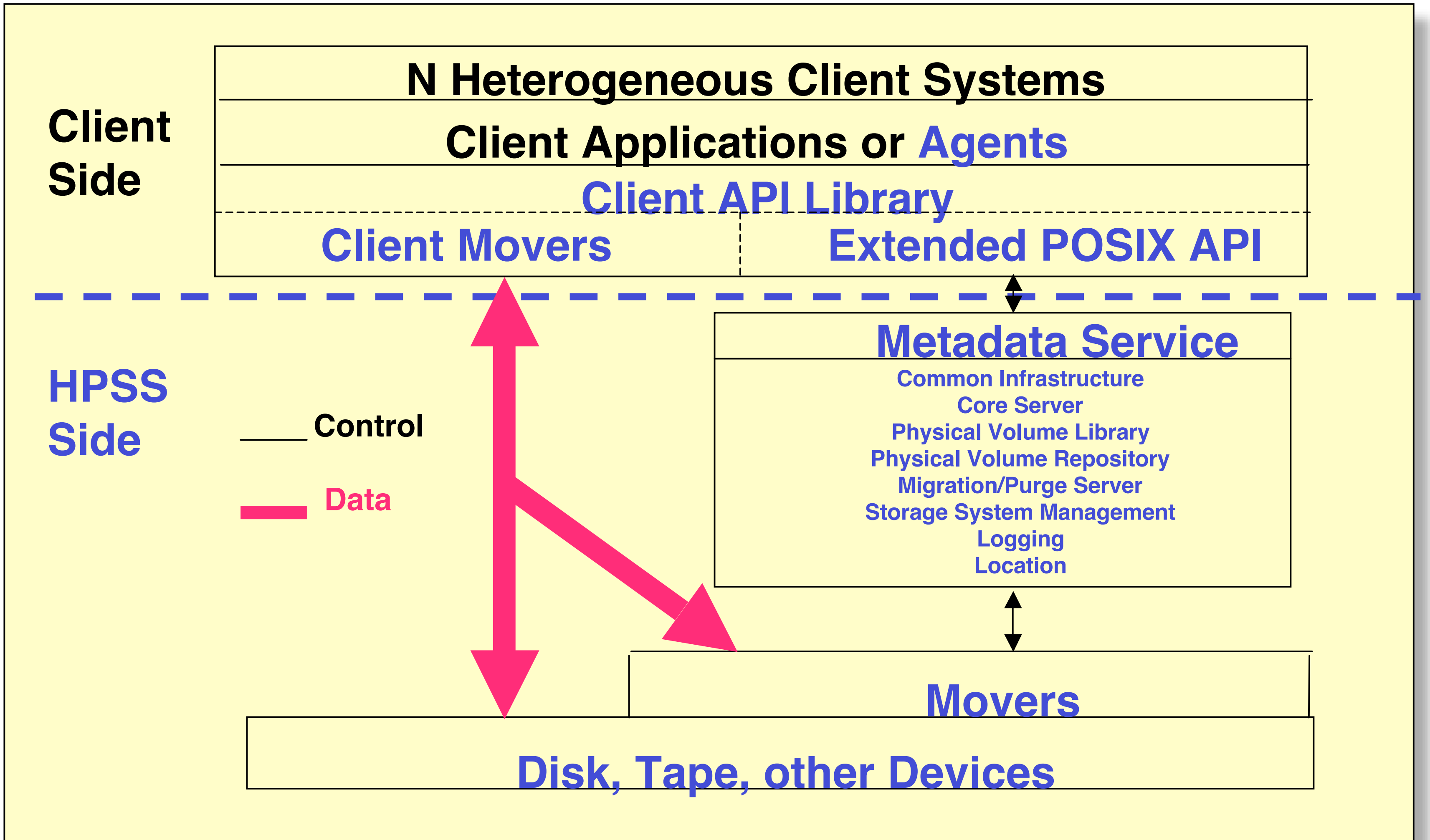
- **HPSS is a true cluster hierarchical storage system**
- **DB2 metadata engine assures reliability and quick recovery**
- **Cluster architecture and metadata architecture support horizontal scaling to:**
  - 10s of petabytes
  - 100s of millions of files
  - gigabytes per second data rates
  - All in a single system
- **Supports technology insertion**
  - Add new components, no need to replace
  - Mix and match vendors and models



# Three factors supporting scalability HPSS

- **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**

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



# HPSS second level architecture and implementation

- **HPSS Infrastructure**
  - Metadata Services (Enterprise class RDBMS (DB2))
    - 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**

# Scalable Robustness

HPSS

- **Architecture**
  - Logically centralized metadata service
  - Separation of metadata storage and user data storage
- **Implementation**
  - Enterprise class RDBMS metadata engine
  - Atomic transactions
  - Log restore time independent of amount of user data
- **Deployment**
  - Mirrored RAID disks backed up at least daily
  - Redundant metadata machine(s) with manual or automatic failover
- **Issues needing work**
  - None identified

# Scalable modular capacity

HPSS

- **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
  - Scalable units
- **Issues needing improvement**
  - None identified

# 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**
  - Improved disk allocation algorithms
  - Improved tape aggregation
  - Improve small file performance (e.g. # of creates/s and read-writes/s)

# Capacity and I/O scaling examples HPSS

- **2.4 PB** Lawrence Livermore National Lab (LLNL) Secure Computing Facility (SCF) (**~34 million files**) **scaled from13 TB** in 1992.
  - **2.7 PB** LLNL Open Computing Facility (OCF) **(~31 million files).**
  - **~1 million directories in the OCF and 1.2 million in the SCF (10K - 90K entries).**
- **5 PB**: Los Alamos National Laboratory (LANL) SCF, **(~ 52M files).**
- **LLNL** - Aggregate data transfer rates to the archive, before HPSS, were well under **10MB/s** and now exceed **2.5GB/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 **50 TB/day**.
- **LANL** - A 2005 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.
- **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.

# HPSS Roadmap to 2011 to Support Petascale Environments

# What do Petascale environments imply for HPSS?

HPSS

- **Relatively few improvements needed.**
  - Need improved small file performance in general and to tape in particular.
  - An improvement in device allocation to minimize I/O and networking conflicts.
  - Improved file system integration.

# 2011 HPSS Performance Requirements

HPSS

| Parameter | 2011 |
|---|---|
| Computing Power as Driver | 10 Petaops |
| Storage Capacity | ~ 25-100 Petabytes (economics main limit) |
| Instantaneous Throughput | ~ 50+ Gigabytes/s (economics main limit) |
| Daily throughput | ~ 250 - 500 Terabytes/day (economics main limit) |
| Small file create-writes/s | Low 1000s (assumes small file aggregation) |

# Determining small file performance requirement difficult: University of Chicago 1 Week BG/L Run

- **The numbers below are only for a 180TF peak partition.**
- **During the week they ran on 32K nodes (32K processors).**
- **They generated 74M files and about 150TB of data.**
- **They generated 32K file filesets organized into a directory for each fileset.**
  - 3.2 10^4 * (200 + 1400 + 700) = 7.4 10^7 files stored in Lustre
- **Here's the number of filesets (each with 32K files) stored in HPSS as single HTAR bundles:**
  - Checkpoint: 200 (Each fileset 640GB), only 15 stored in HPSS
  - Particle Plotfile: 1400 (Each fileset 20GB), all stored in HPSS
  - Grid Plotfile: 700 (Each fileset 0.5GB), all stored in HPSS
- **HTAR reduced the number of objects to be managed by HPSS by factor of 10,000.**
  - Even with such aggregation we assume could need small file transactions/s in the range of low 1000s/s for a petascale environment.

# HPSS Roadmap to 2011- Focus: Scaling Small File Performance

HPSS

**Deploy 6.2**

**Q1 2006**
**6.2**
**Complete**
**Core Server**
**Infrastructure**
**Upgrade**

**Deploy 7.1**

**Q3 2007**
**7.1**
**Performance Tuning**
**Logically Single**
**Core Server on Cluster**
**Goal - 5X performance improvement**

**Deploy 8.1**

**Q1 2009**
**8.1**
**Ver. 1 Dynamic Multiple Core Server Clusters**
**Ver. 1 (OSD support?*)**
**Ongoing Tuning**
**Goal >10X improvement**

**Deploy 8.2**

**Q4 2010**
**8.2**
**Ver. 2 Dynamic Multiple Core Server Clusters**
**Ver. 2 (OSD support?*)**
**Ongoing Tuning**
**Goal >5X improvement**

---

***Need to establish requirement and economics but straightforward to support**

# HPSS near term requirements
## (Release 7.1, Q3 2007)

- **Improve performance (goal 5X)**
  - Improve small file performance (e.g. improve tape file management, improve metadata performance overall)
  - Facilitate greater throughput (e.g. Storage Server device allocation algorithm, above)
- **Improve site integration**
  - File system integration (e.g. Lustre, Panasas, GPFS, VFS)
    - GPFS <-> HPSS demonstrated at SC 05
  - Mover device affinity (multiple Movers can share a device, clients)
  - 64 bit PFTP
- **Improve transparency and administration**
  - Dynamic segment size allocation
  - Multiple streams of COS changes
- **Provide a common Trilab user interface (in planning phase during this timeframe)**

# Beyond 7.1 - Key requirement is continued improvement in metadata performance

HPSS

- **To further improve metadata performance requires more metadata handling parallelism.**
- **There are three main areas being studied:**
  - Multiple Core Servers with dynamic load balancing
  - More intelligent devices (e.g. OSDs)
  - More processing, aggregation and caching in clients.

# Multiple dynamic Core Servers

- **Ultimately achieving more parallelism beyond multithreading is required for more scalable metadata performance.**
- **Currently have multiple subsystems (Core Servers) based on static name space assignment distribution.**
  - Multiple subsystems load balance by static name space subtree allocation.
- **HPSS project currently studying how to most effectively utilize DB2 partitioning and other capabilities to support multiple dynamic Core Servers (Metadata Servers).**

# Object Storage Device support straightforward

- **OSDs offer another way to improve parallelism at a lower level.**

- **Basic integration is straightforward**
  - HPSS architecture has separation of data and control logic and appropriate object abstraction layering supporting a segment abstract object.
  - OSD support would integrate simply into segment layering of the Storage Server.
  - Need to modify authentication so OSD can authenticate capabilities for each I/O.
  - Client library and Mover logic needs adaptation as Client will do direct I/O on cached metadata from Open.

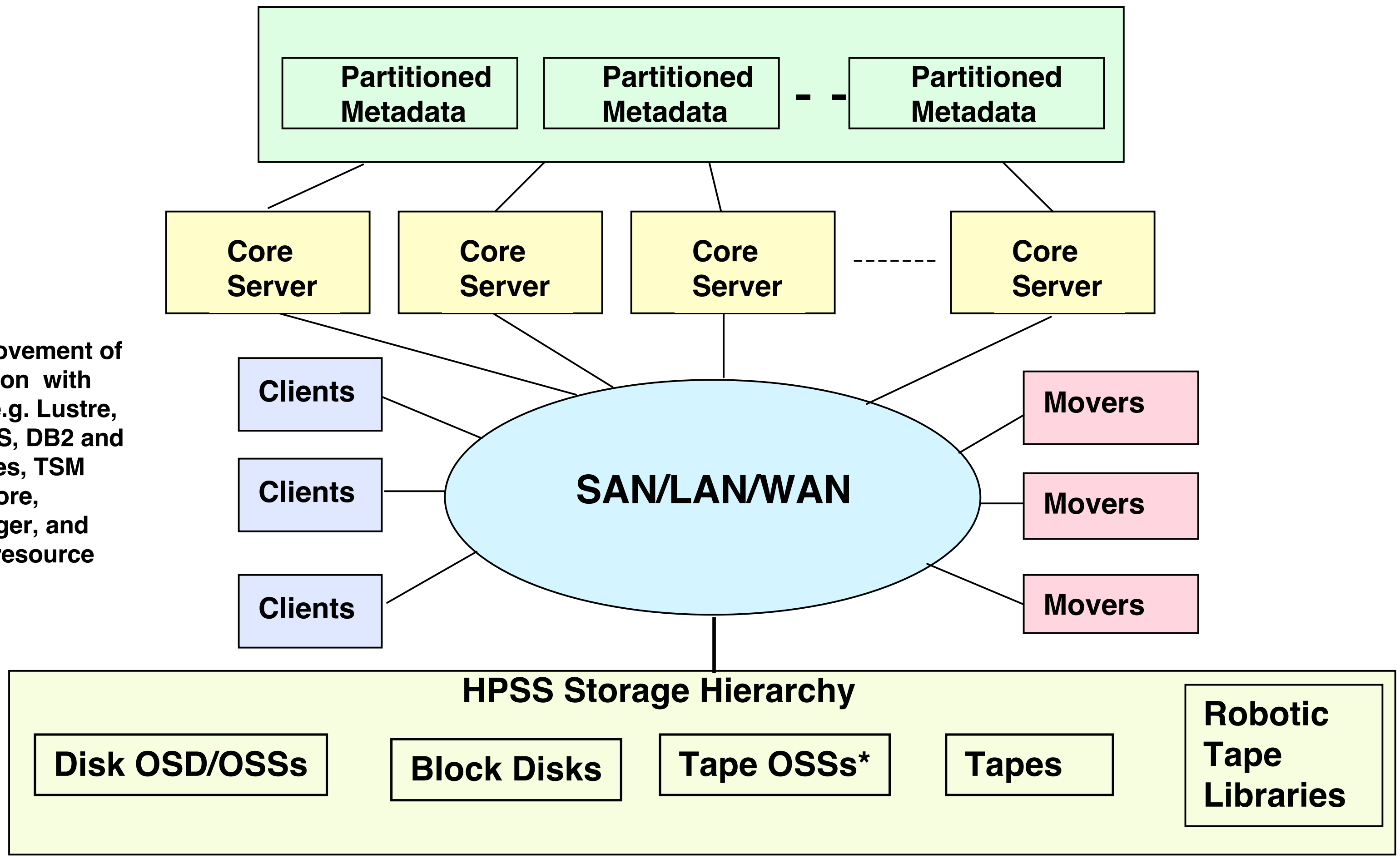
- **Questions**
  - What percent of current operation time(e.g. create, read, write) would be in this level of metadata processing and thus how much would system performance benefit?
  - What Mover latency would be saved, again how much would performance benefit?
  - How to assure metadata in OSD/OSSs is “safe”? (There are issues with current implementations and deployments)
  - Would developing tape OSSs make sense given all the latencies and other issues managing tape?

# What things might make sense for an HSM to do in the client?

- **Currently HPSS client utilities such as PFTP, PSI, HSI, HTAR do sophisticated client side operations to optimize performance.**
  - Examples include file bundling (aggregation); data transfer, striping, device, multithreading, staging optimization; restart and error recovery; directory listing caching;  and more.
- **Studying other client level functionality to increase parallelism and latency hiding.**
  - The standard system approaches are forms of buffering, aggregation and caching.
    - One example planned is transparent Client access to bundled file metadata.
  - Given the requirement for very high robustness, which options make sense for an HSM/archive?

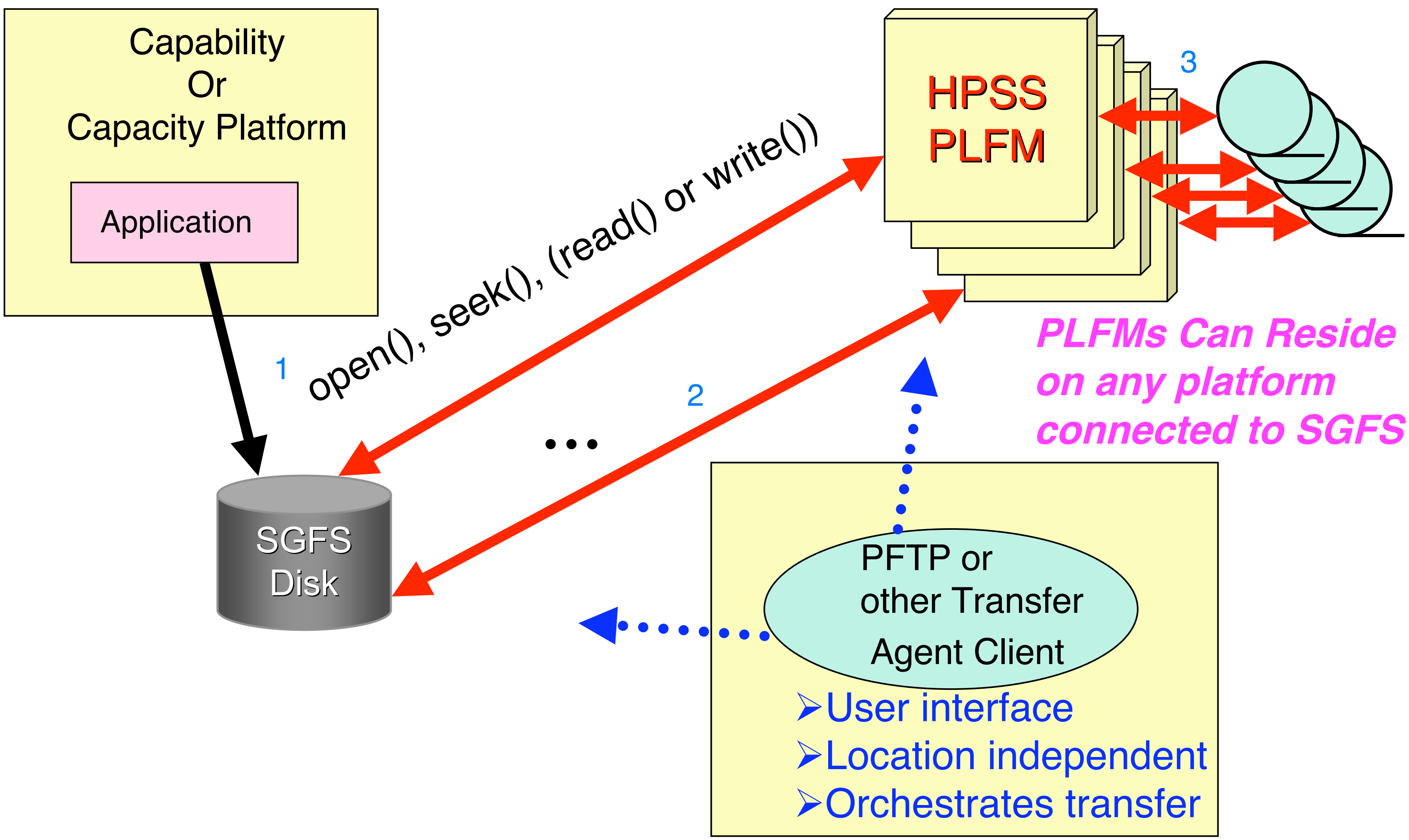
# HPSS (8.1, 8.2) in 2009 - 2011

Ongoing improvement of client integration with file systems (e.g. Lustre, Panasas, GPFS, DB2 and other databases, TSM backup & restore, Content Manager, and Grid storage resource brokers)




* No such devices currently exist or are planned nor is it clear that such devices make economic sense. I'm very interested in discussion here.

# Example: Integrating HPSS with a Scalable Global FS (SGFS) using Parallel Local File Movers (PLFM)



# Yes Virginia, There is an HPSS in Your Future

- **HSMs/Archives are required as far into the future of storage devices as we can see.**
- **HPSS has demonstrated significant scaling capabilities:**
  - **100** for capacity to petabytes,
  - **1000** for single file bandwidth to GB/s.
  - **1000** for instantaneous throughput to GB/s,
  - **1000** for daily throughput to 10s TB/day, and
- **The object-oriented, flexible, network-centric architecture of HPSS and modular industry standard product infrastructure are sound.**
  - Use of an enterprise class DB engine is crucial part of scalability strategy
  - Enterprise DB also key part of HPSS robustness strategy.
- **The HPSS architecture and implementation have lots of room for further scaling in I/O, capacity, metadata performance and other dimensions by further orders of magnitude in the future.**
  - Multiple Core Servers, OSDs, more client side functionality fit naturally.
- **HPSS has roadmap to meet future Petascale environment HSM/Archive requirements with relatively few changes.**

**Acknowledgement**

I wish to thank the many developers within the HPSS Collaboration who have created HPSS. This work was, in part, performed by the Lawrence Livermore National Laboratory, Los Alamos National Laboratory, Oak Ridge National Laboratory, National Energy Research Scientific Computing Center and Sandia National Laboratories under auspices of the U.S. Department of Energy, and by IBM Global Services – Federal.

**Disclaimer**