HPSS/DFS: Integration of a Distributed File System with a Mass Storage System

NASA Goddard Conference on Mass Storage Systems and Technologies

IEEE Symposium on Mass Storage Systems

March 23-26, 1998

Rajesh Agarwalla
Transarc Corporation

Rena Haynes
Sandia National Laboratories

The Team

IBM

TRANSARC

Lawrence Livermore National Laboratory

HPSS
High Performance Storage System

ornl

Sandia National Laboratories

Los Alamos

This work was performed jointly under the auspices of the U.S. Department of Energy as part of the Accelerated Strategic Computing Initiative (ASCI) by LLNL (Contract W-7405-ENG-48), LANL (Contract W-7405-ENG-36), SNL (Contract DE-AC04-94AL85000).

Motivation

- Information intensive era
- Cost effective storage of data
- Efficient and seamless access to data

Storage of data

Memory hiearchy

Primary	RAM	60ns	$2 per MB
Secondary	Disk	8ms, ~40MB/s	$0.10 per MB
Tertiary	Tape	>4 min, ~5MB/s	$0.002 per MB

⟹ **Mass storage systems**

Access to data

- Integrated access
- Data integrity
- Security

→ File system

- Scalable across geographically disparate locations

→ Distributed file system

- High speed I/O

→ Parallel paths
Third party transfers

Solution

- Integrate filesystem with mass storage system
- Migrate data from filesystem to mass storage system
- Cache data from mass storage system to filesystem
- Transparent migration / caching
- Efficient migration/caching mechanism

Previous approaches

- Filesystem uses mass storage system at backend
  - DMF, AMASS
  - Needs kernel modifications with OS upgrades by mass storage system vendor

- Mass storage system implements a filesystem interface
  - CFS, Unitree, HPSS
  - Lacks benefits of distributed filesystems
  - May need specialized clients

Our approach

- DMAPI
  - Recent standard interface between filesystem and data management apps
  - Adopted by X/Open - XDSM

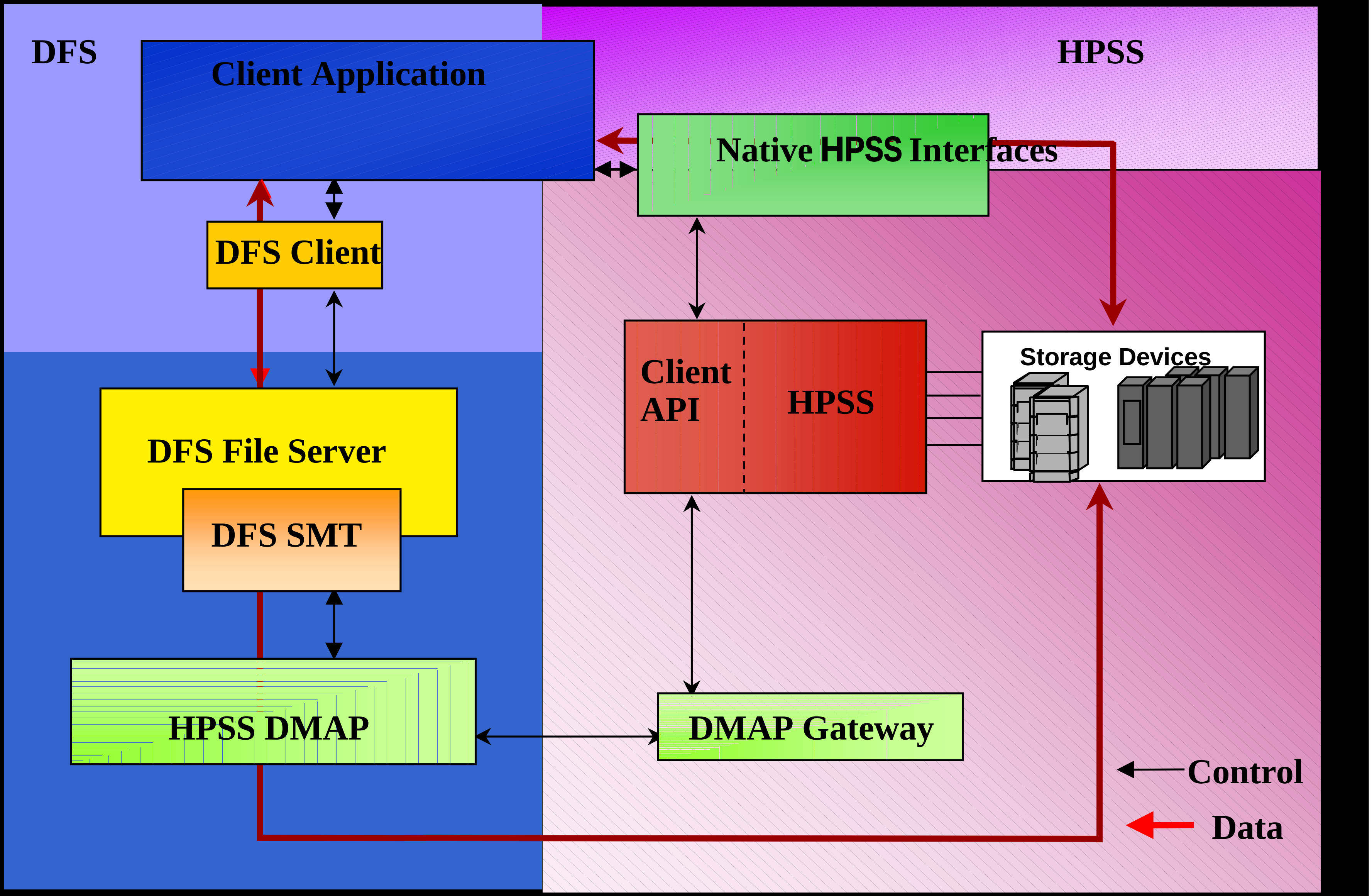
- Integrate
  - DFS ^{TM} distributed file system with
  - HPSS mass storage system
  - via DMAPI

- DFS Storage Management Toolkit (DFS SMT) layer
  - An implementation of the DMAPI standard for DFS
  - HPSS interface layer using DFS SMT

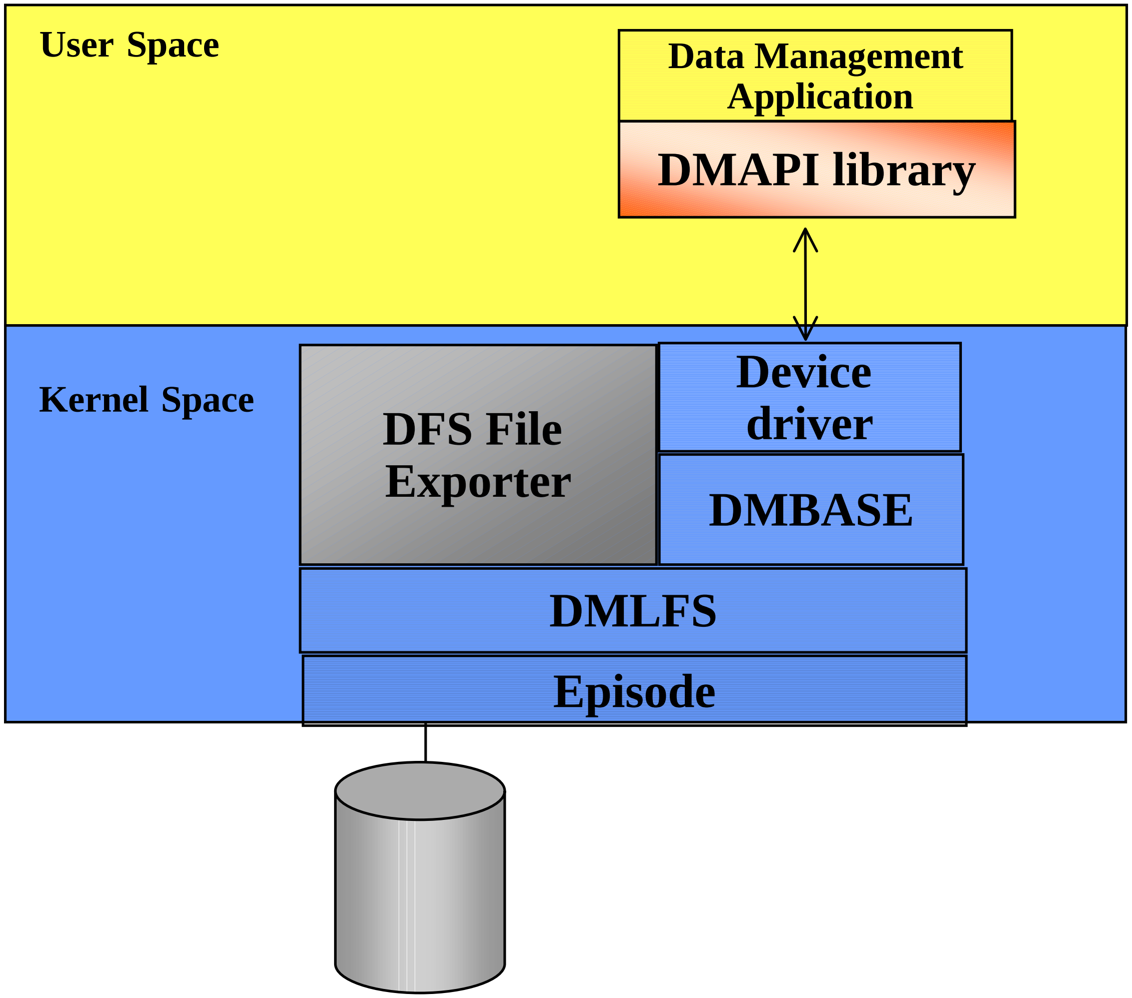
Our Requirements

- **Transparent archiving and caching of data**
- **Partial file residency**
- **No kernel mods by mass storage vendor across OS updates**
- **Preserve existing functionality/performance of DFS and HPSS**
    - **Add a mode where file modifications are visible in both DFS and HPSS**

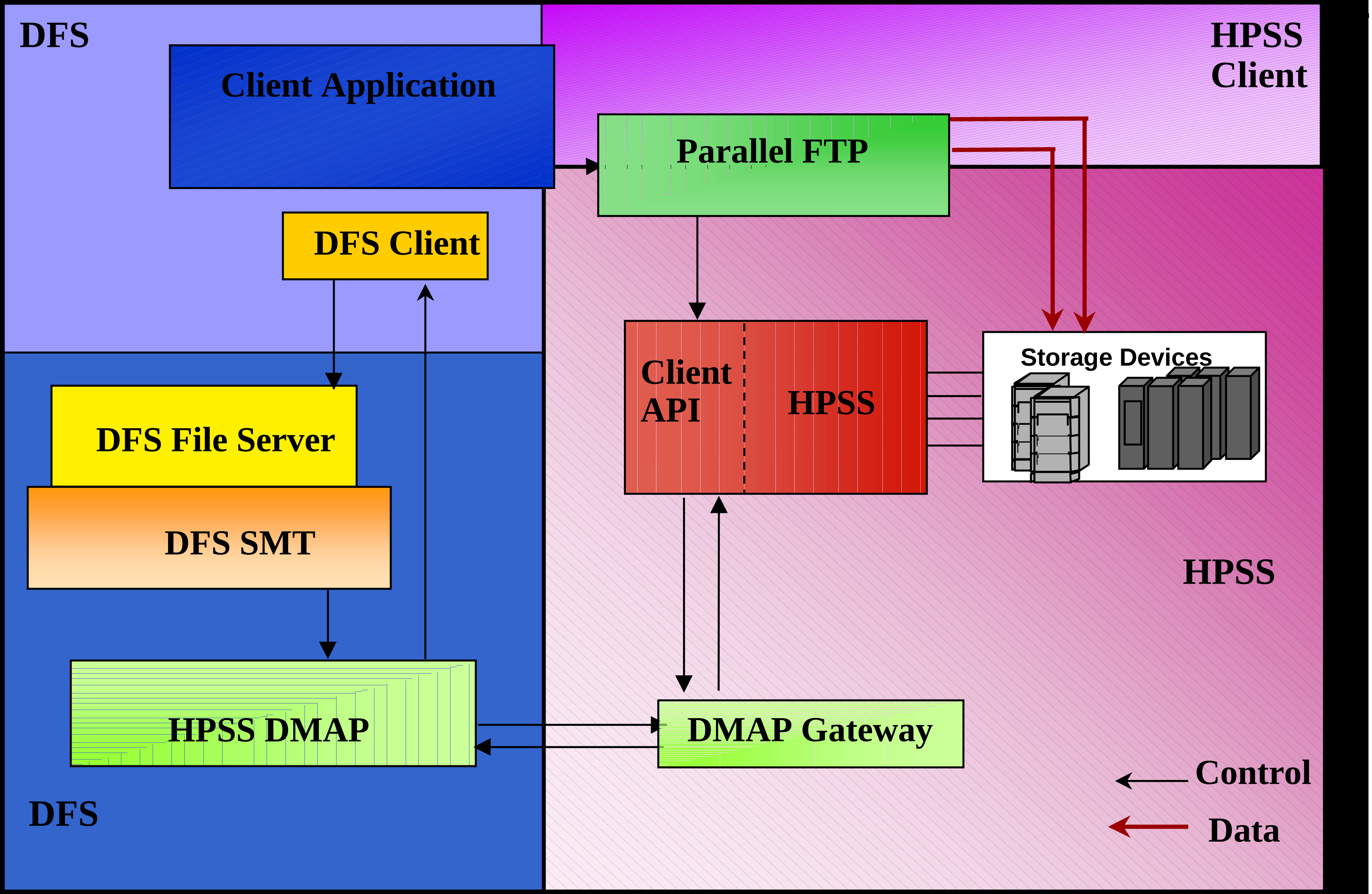
Integrated HPSS-DFS Architecture



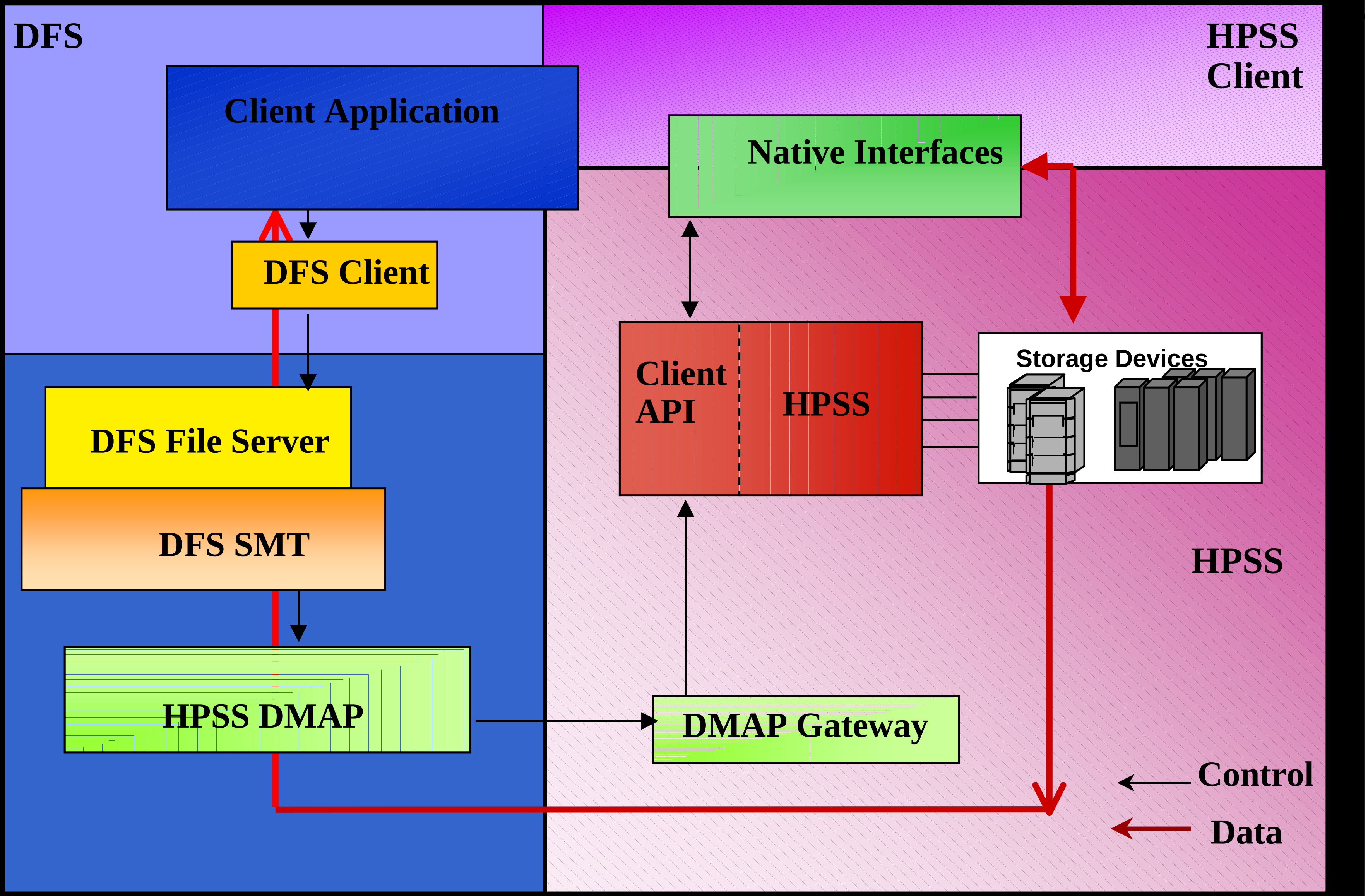
DFS Storage Management Toolkit (SMT) Architecture



HPSS Create Example



DFS Read Example



DFS SMT Features

- Filesystem sends notifications to DM application
- Then if necessary waits for response from DM application
- DM application processes notification
  - e.g. caches data into the filesystem from mass store
- DM application responds to the notification
- DM application can initiate operations on files via DMAPI
  - e.g. migrating data from filesystem to mass store
  - e.g. making migrated data non-resident in the filesystem

DFS SMT Features - 2

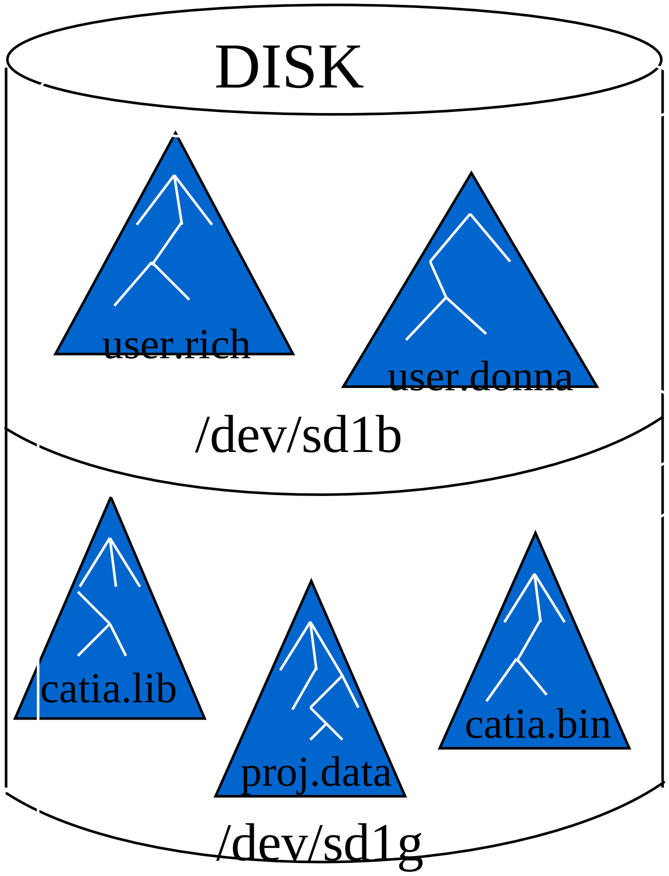
- Provides for storage of DM attributes with files
    - Filesystem visible attributes
        - e.g. filesystem operations which should generate notifications to DM app
    - Filesystem opaque attributes
        - Understood by DM application
        - e.g. pointers to migrated data
- Implements all required DMAPI features

DFS SMT Features - 3

- Many optional DMAPI features provided
    - Persistent event masks
    - Persistent managed regions
    - Persistent attributes
    - Real removal of residency of migrated data
    - Punch holes in files
- Non-blocking lock upgrades

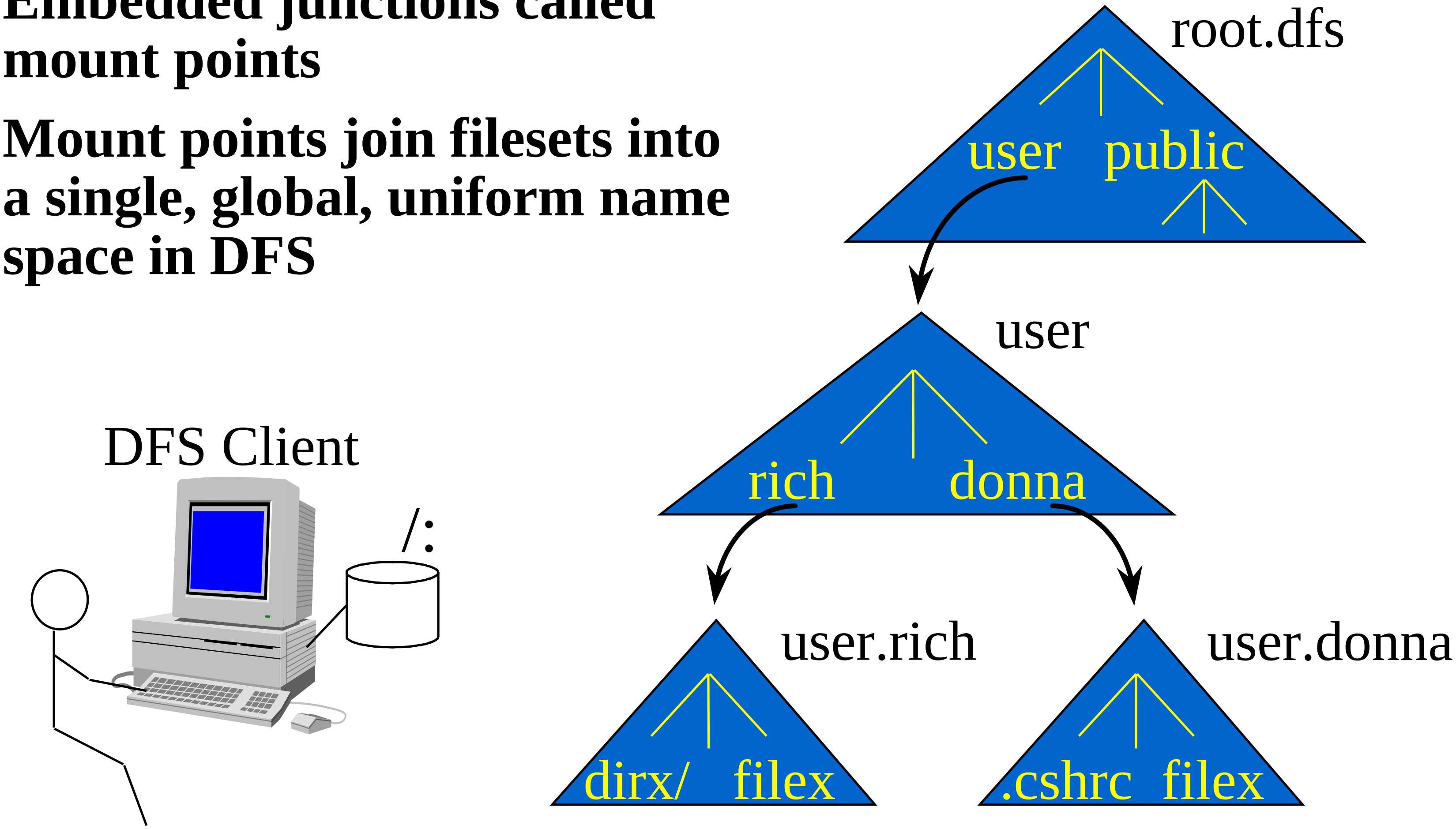
Concept - DFS Filesets

- Total file space at fileservers divided into *filesets*
- Each fileset is a separate tree-like filesystem
- Fileset: collection of related files
- Unit of administration, backup, replication



How are filesets linked in DFS?

- **Embedded junctions called mount points**
- **Mount points join filesets into a single, global, uniform name space in DFS**



Other DFS Additions

- Episode
  - Support for punching holes in files
    - Ability to mark holes as *offline* data for purged data
  - Support for storing file attributes for files
    - Attributes inherently linked with respective file (No auxiliary file)
- Data backup
  - Filesets are unit of data dump and restore
  - Dump/restore facilities extended for file attributes, purged holes
  - Migrated and purged data not recalled currently when dumping

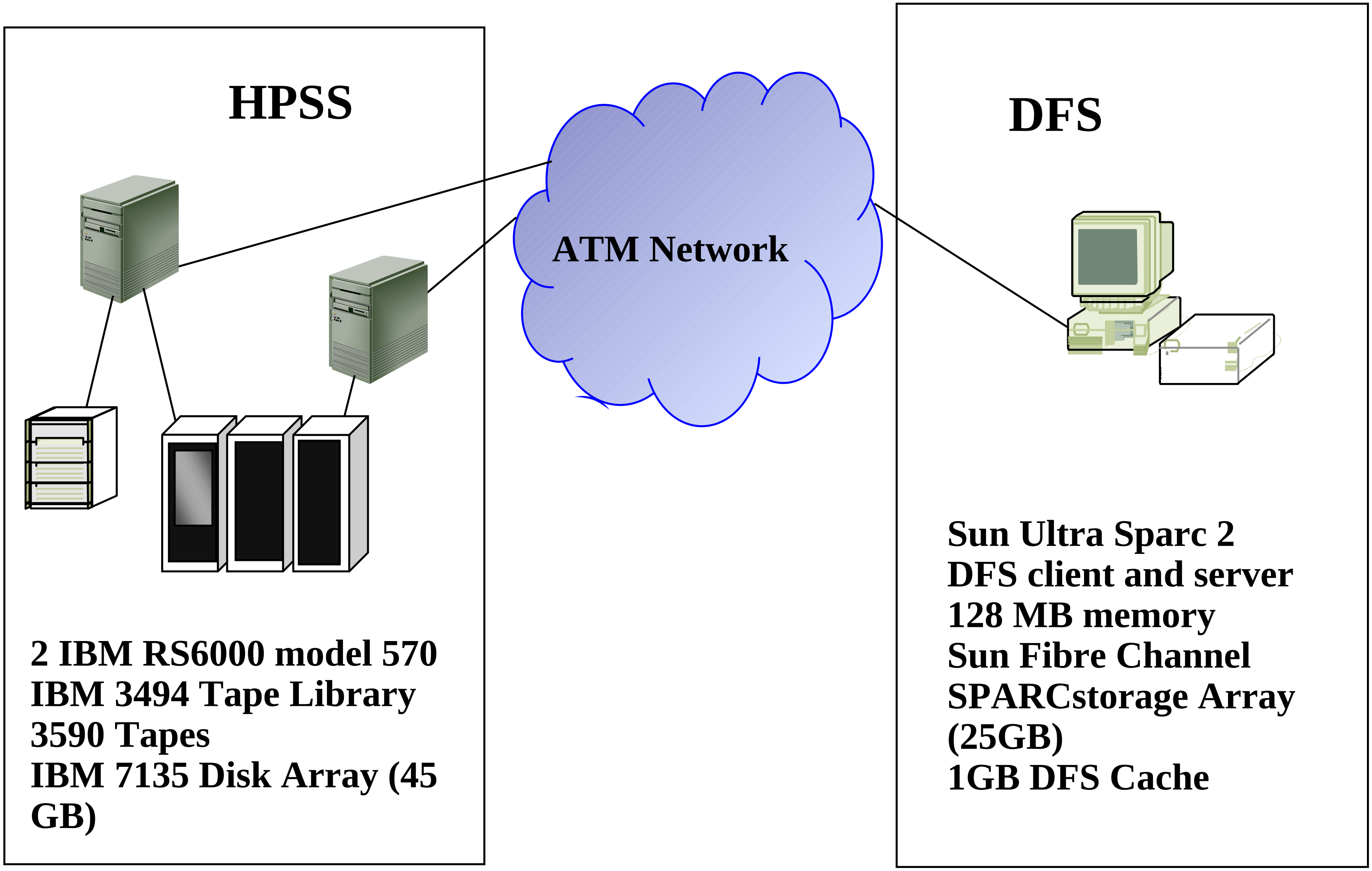
DMAPI Extensions for DFS SMT

- Filesets and aggregates
  - mounting/unmounting aggregates
  - fileset destruction
  - enumerate fileset information
- Management interfaces
  - Scan by attribute
  - DCE security authentication information
  - ACL / permission events
- Mirrored fileset support
  - Synchronous post events for name space modification
  - Events for permissions changes
  - Association of pre and post events

HPSS Additions

- Data residency supported in HPSS
  - HPSS Only
  - Archived
  - Mirrored
- Name Server
  - Fileset type
  - Junctions to link filesets into HPSS tree
- Client API
  - Fileset behavior
  - Junction processing
  - Shared transactional boundaries to support atomic behavior between DFS and HPSS
- Bitfile Server - data consistency
- File families

Performance Test Hardware Configuration at Sandia National Laboratories



HPSS

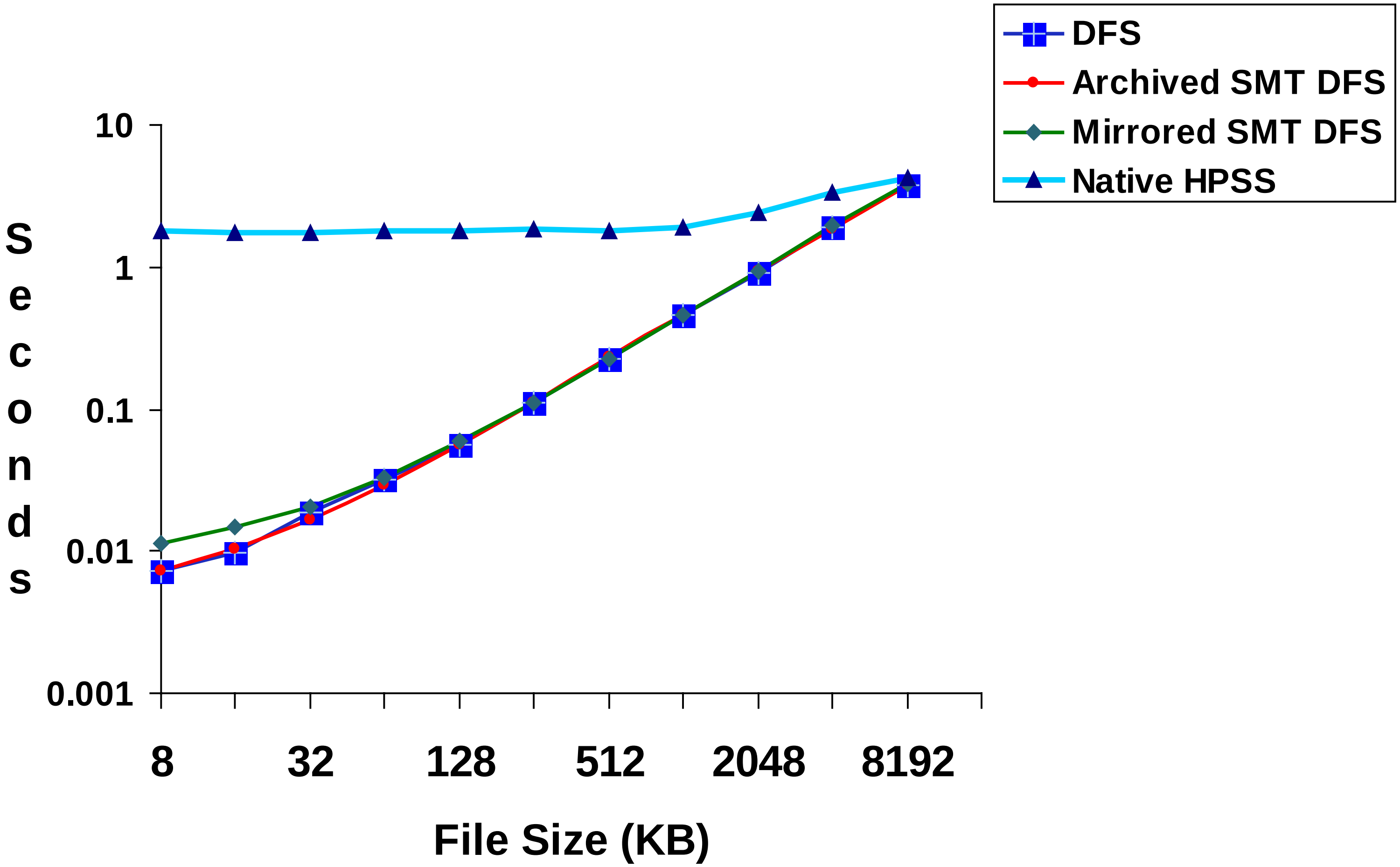
ATM Network

DFS

2 IBM RS6000 model 570
IBM 3494 Tape Library
3590 Tapes
IBM 7135 Disk Array (45 GB)

Sun Ultra Sparc 2
DFS client and server
128 MB memory
Sun Fibre Channel
SPARCstorage Array
(25GB)
1GB DFS Cache

Average Time to Write a File



Connectathon Test Description

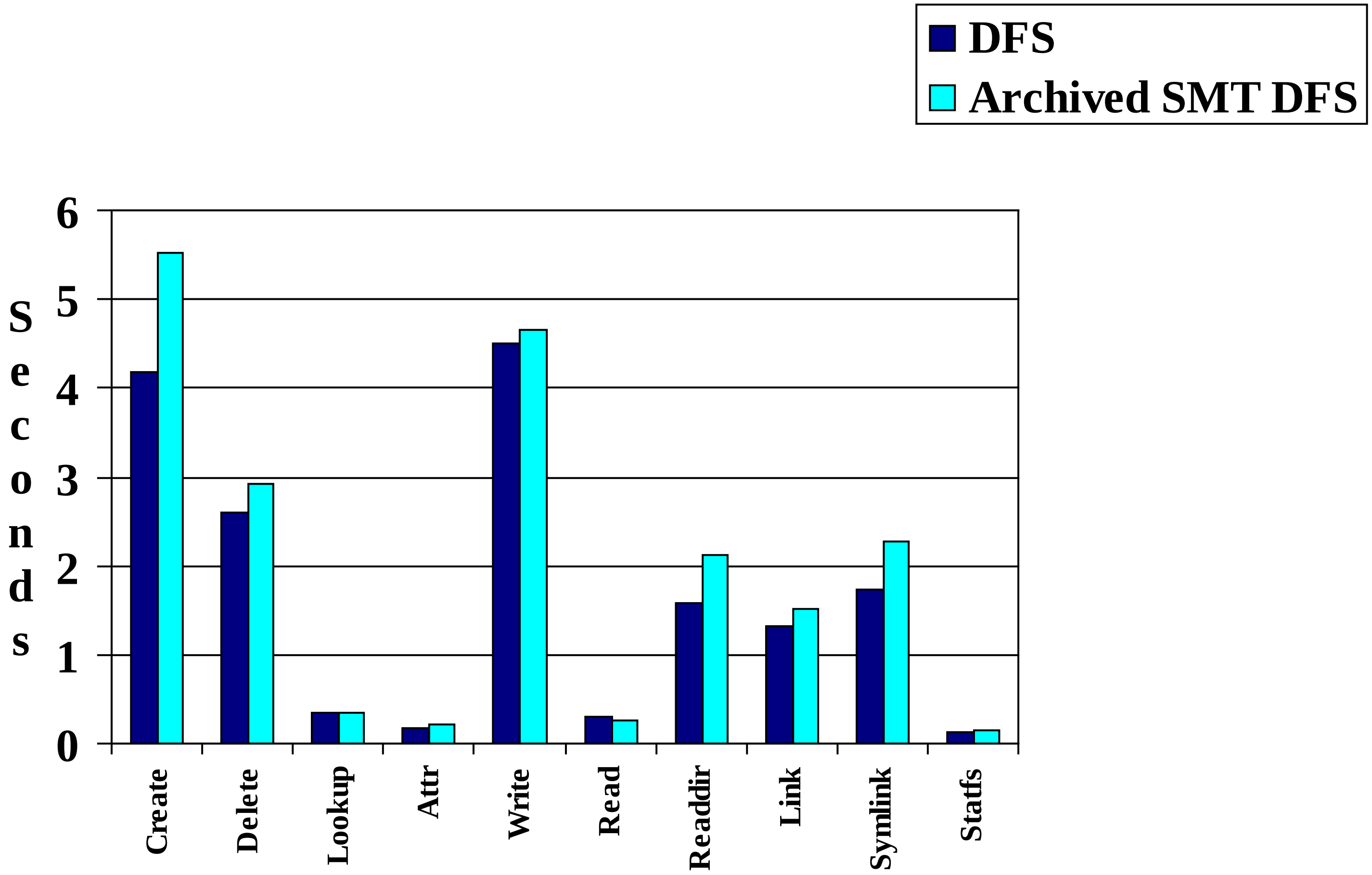
- **Create : create 155 files 62 directories 5 levels deep**
- **Delete: remove 155 files 62 directories 5 levels deep**

- **Lookup: 500 getwd and stat calls**
- **Attr: 1000 chmods and stats on 10 files**
- **Write: write 1MB file 10 times**
- **Read: read 1MB file 10 times**

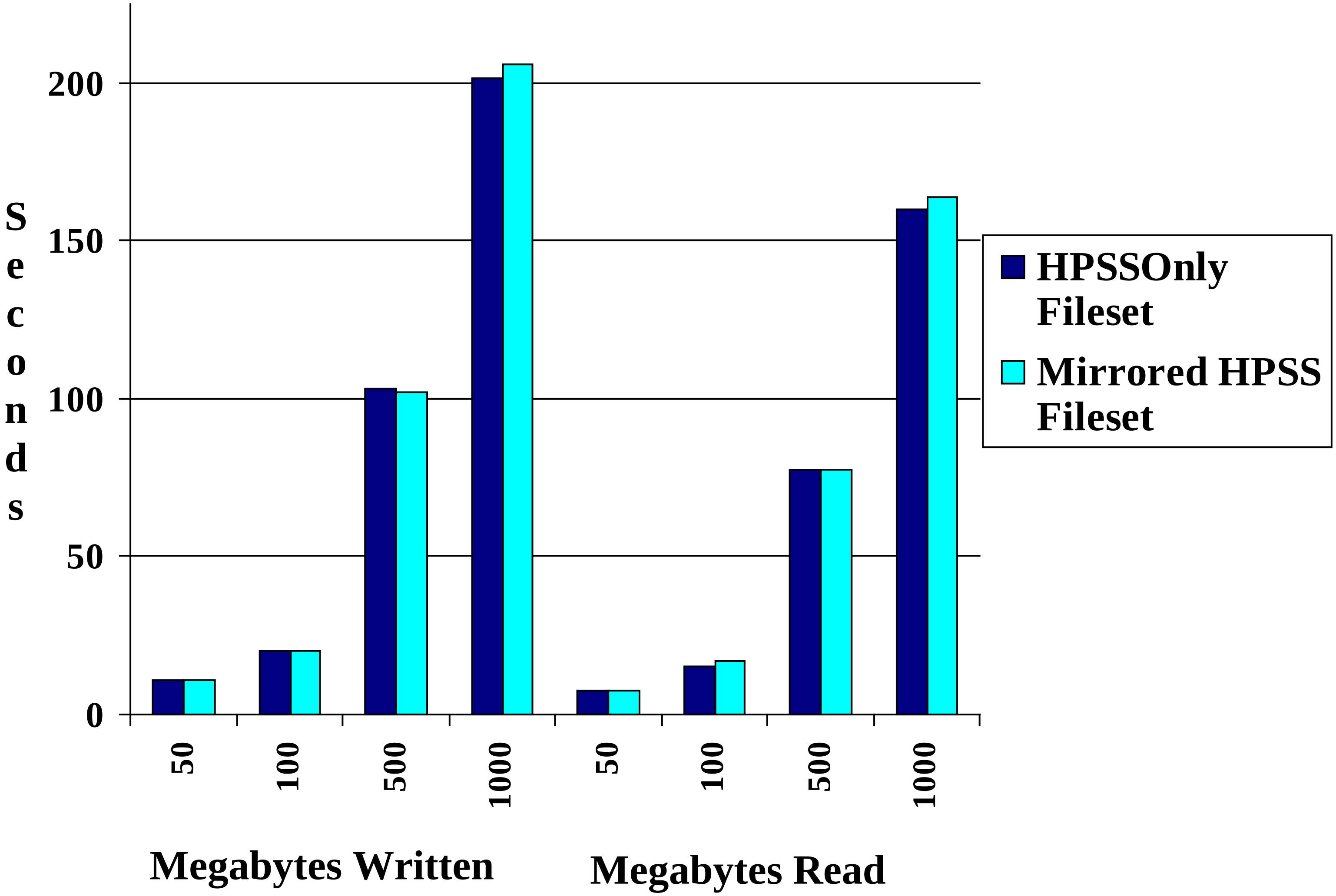
- **Readdir: 20500 entries read, 200 files**
- **Link: 200 renames and links on 10 files**
- **Symlink: 400 symlinks and readlinks on 10 files**

- **Statfs: 1500 statfs calls**

Integrated DFS Connectathon Performance



Integrated HPSS I/O Performance



Conclusions

- Flexible DFS/HPSS Integration

- Minimal impact on data I/O rates
  - Archived DFS file creation equivalent to DFS at 8KB
  - Mirrored DFS file creation equivalent to DFS at 32KB
  - Mirrored HPSS performance equivalent to native HPSS (<3% difference)

- Connectathon performance overhead
  - archived DFS connectathon performance overhead ~20%

- Greater administrative complexity

Future Work

- **DFS**
  - **Support for fileset movement and replication**
  - **Support for full fileset dumps**
  - **Client visible DM attributes**
- **HPSS DMAP ports to other platforms**
- **Easier administration tools**
- **Performance enhancements**
- **DMAPI extensions**
  - **better support for distributed systems**
  - **name space synchronization**
  - **parallel file system support**

Additional Information

- **HPSS URL:**

  www5.clearlake.ibm.com:6001

- **DFS URL:**

  www.transarc.com

- **Availability:**

  Sun Solaris and IBM AIX platforms

  July - September