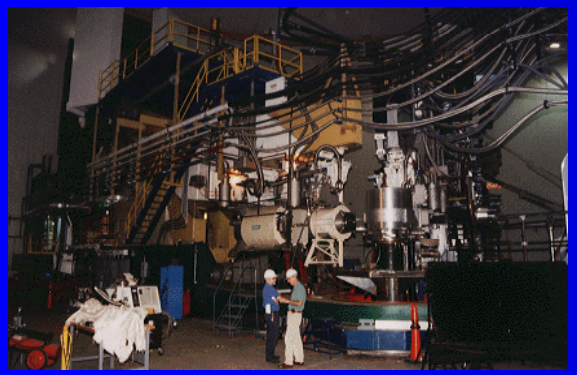
Building the Mass Storage System at Jefferson Lab

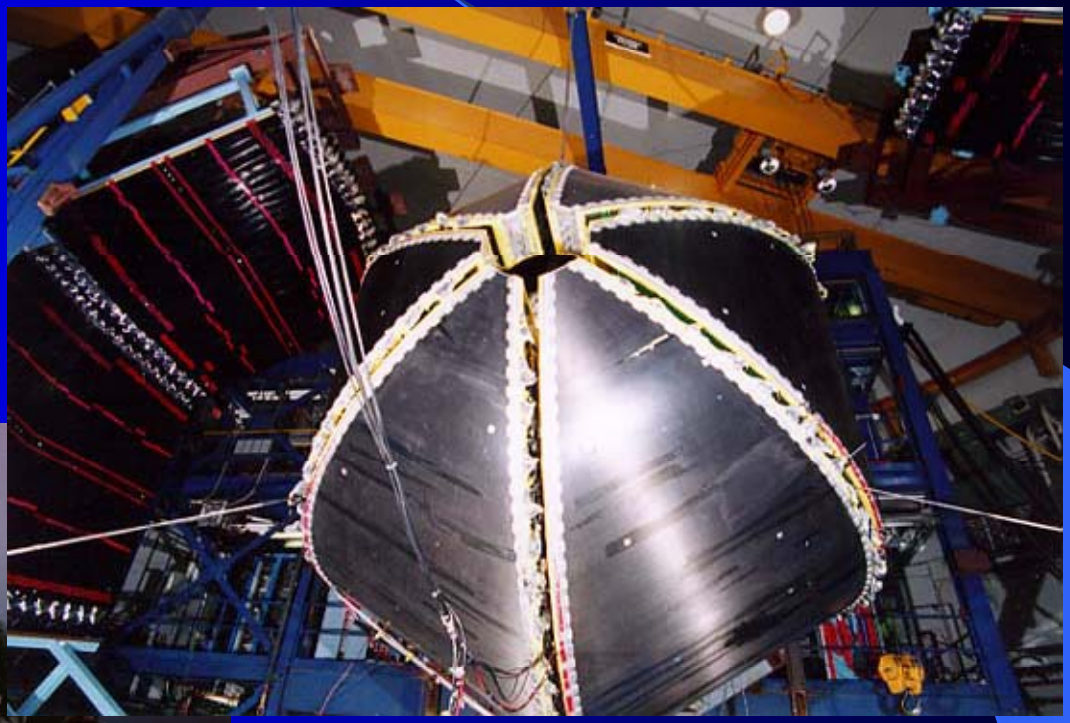
Andy Kowalski
Ian Bird, Bryan Hess
SURA/Jefferson Lab

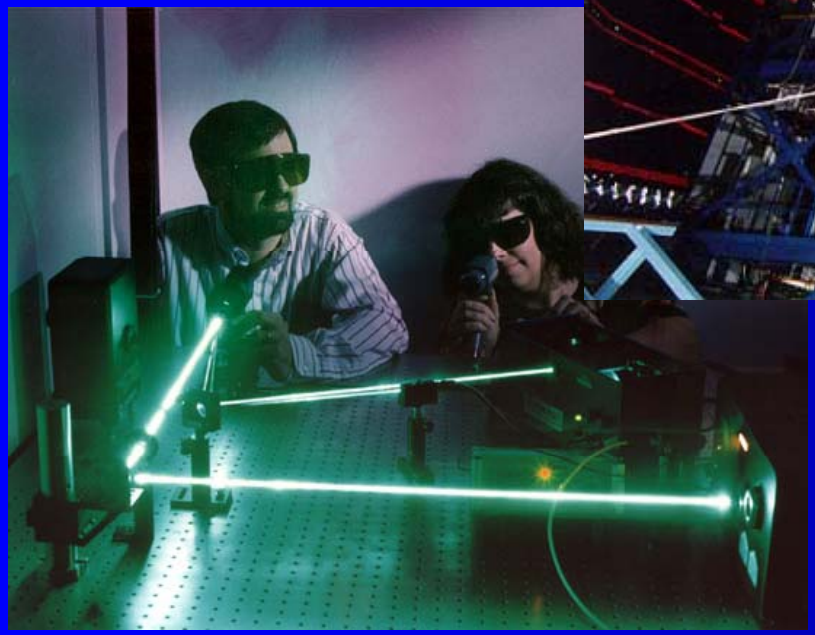
Jefferson Lab

- Who are we?
  - Thomas Jefferson National Accelerator Facility
  - SURA/DOE
- What do we do?
  - High Energy Nuclear Physics
  - Operate a 6.07 GeV continuous electron beam accelerator
  - Free-Electron Laser (1720 Watts)
- Research
  - quark and gluon

Jefferson Lab





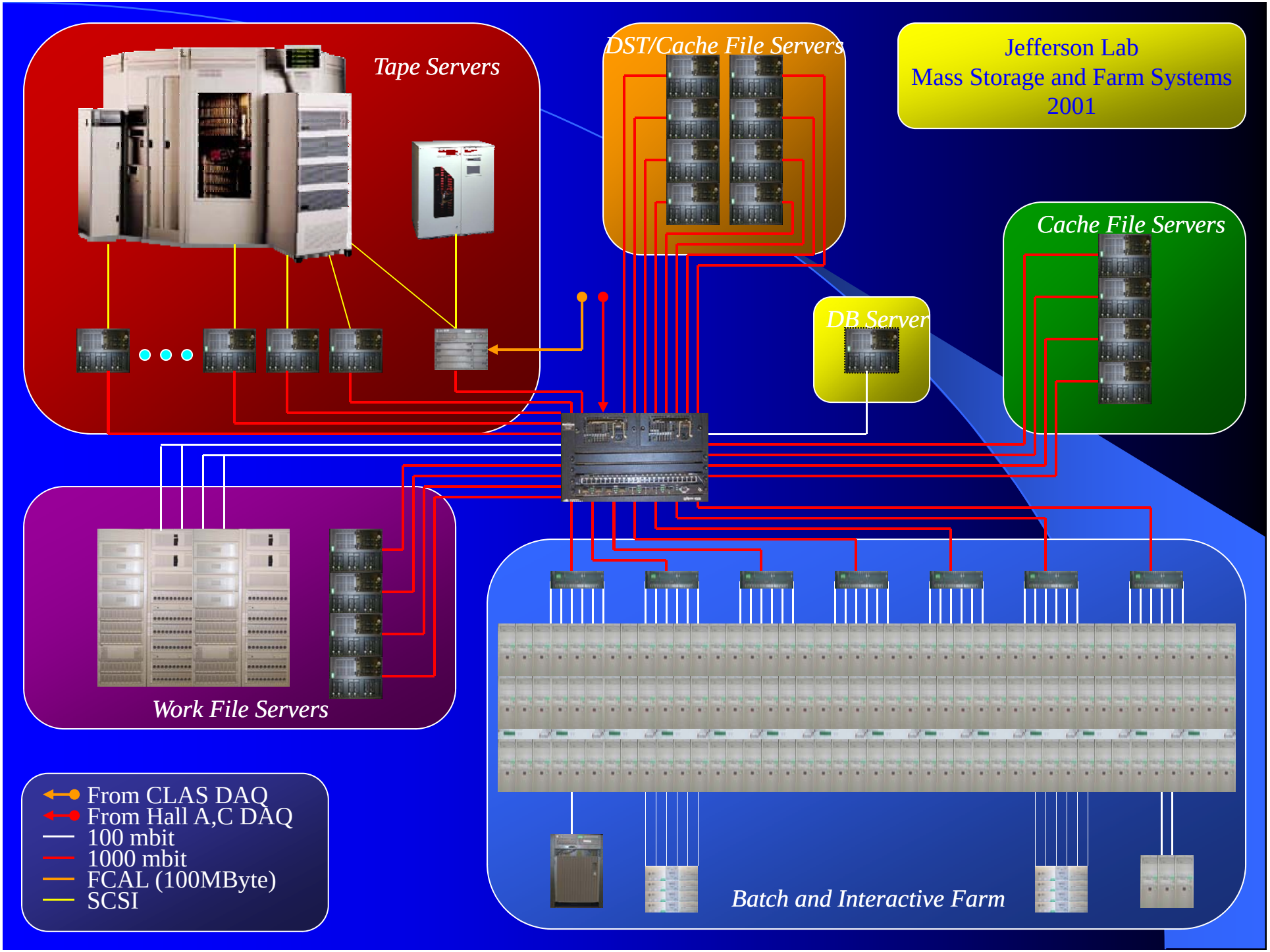


Environment

- Three experimental halls
  - Data rates
    - 1.5 TB/day, 1-100 GB/day, 1-100 GB/day
    - total I/O rate over 2TB/day with batch farm
- Storage Capacities
  - STK Powderhorn 9310 SILO
    - 8 - SD3 (Redwood) tape drives
    - 10 - 9840 tape drives
    - 5 - 9940 tape drives
  - Disk Space 26TB of RAID
    - 2430GB – Stage Areas
    - 15515GB – Cache Areas
    - 8792GB – User Work Areas

Environment Cont.

- Gigabit and Fast Ethernet
- Fiber Channel
- Batch Farm 6172.6 SPECint95
  - 4 Dual Sun Ultra2
  - 125 Dual Pentium II,III (Linux)
- Analysis Farm 200+ SPECint95
- Load Sharing Facility (LSF)
- JLab Asynchronous Storage Manager (JASMine)


Jefferson Lab
Mass Storage and Farm Systems
2001
Tape Servers
DST/Cache File Servers
Cache File Servers
DB Server
Work File Servers
Batch and Interactive Farm
From CLAS DAQ
From Hall A,C DAQ
100 mbit
1000 mbit
FCAL (100MByte)
SCSI

Before JASMine

- Open Storage Manager
  - Computer Associates dropped support January 2000
  - Not distributed
  - Two installations – mss1 and mss2
- TapeServer Front-End
  - Written in Java
  - Makes two OSM servers act as one
  - Stages data to/from disk from/to tape
  - Provides a user interface

JASMine

- JASMine
  - Replacement for OSM
  - Distributed Data Movers and Cache Managers
  - Scalable to the needs of the experiments
  - Smart scheduling
  - Off-site cache or ftp servers for data exporting
- JASMine Cache Software
  - GRID Aware – PPDG

Architecture: Software

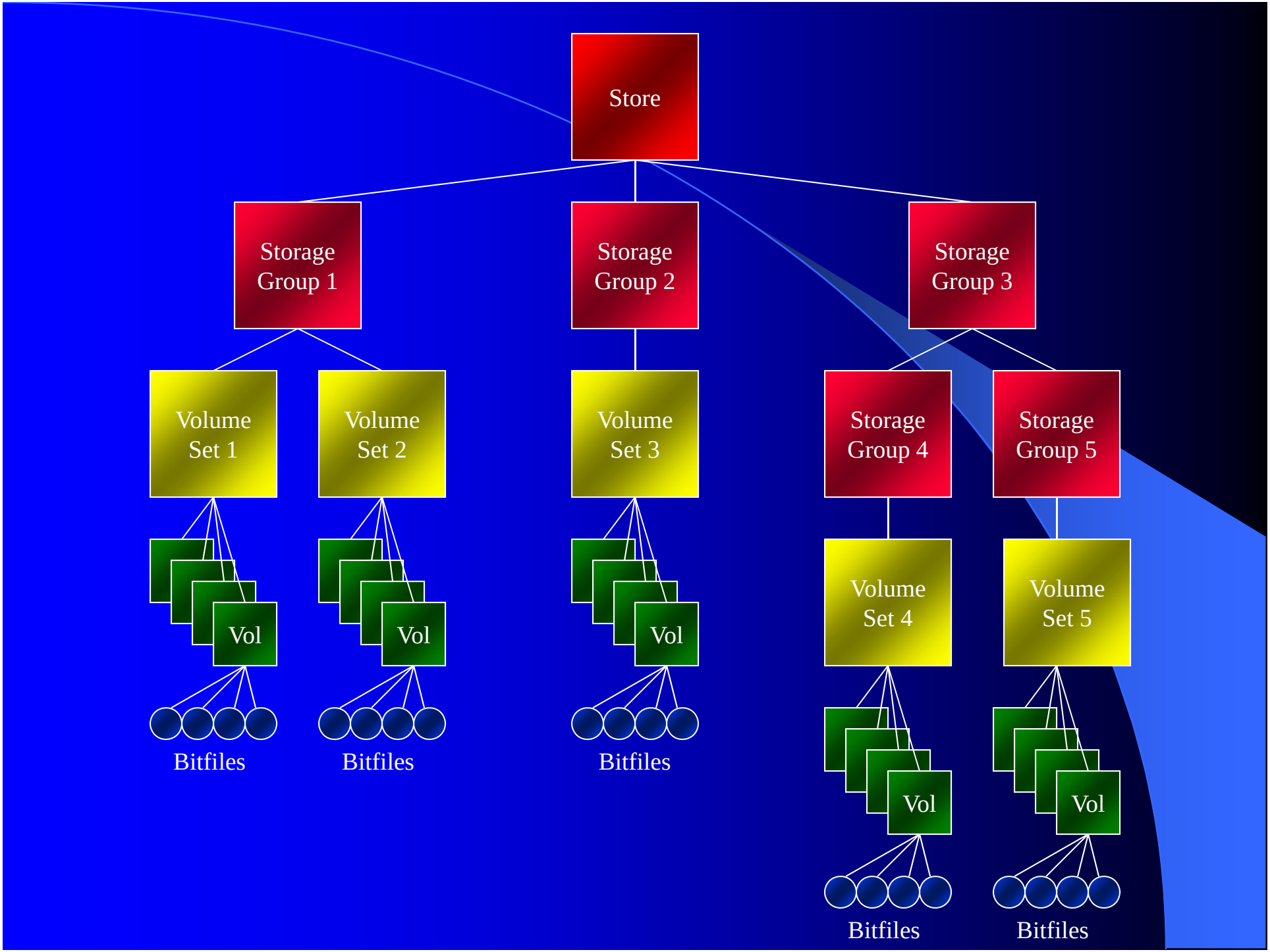
- Java 1.3
  - For data movement, as fast as C code.
  - Fast, easy development.
  - Does garbage collection.
  - Runs everywhere, no porting.
  - JDBC makes using and changing databases easy.
- Cache manager runs on each cache server.
  - Hardware is not an issue.
  - Need a JVM, network, and a disk to store files.

Software cont.

- MySQL database used by all servers.
  - Fast and reliable.
  - SQL
- Tape Labels
  - ANSI standard with extra information
  - OSM support to read legacy tapes
- Protocol for file transfers
- Writes to cache are never NFS
- Reads from cache may be NFS

JASMine Logical Storage Organization

- Store
  - A logical entity made up of libraries, servers, data movers, and a database.
- Storage Group
  - An object that belongs to a store and is itself a collection of storage groups or volume sets.
- Volume Set
  - An object that belongs to a storage group and is itself a collection of volumes.
- Volume
  - A unit of storage media.
- Bitfile
  - The copy of a file that has been copied into a store.


Store
Storage Group 1
Storage Group 2
Storage Group 3
Volume Set 1
Volume Set 2
Volume Set 3
Storage Group 4
Storage Group 5
Vol
Vol
Vol
Volume Set 4
Volume Set 5
Vol
Vol
Bitfiles
Bitfiles
Bitfiles
Bitfiles
Bitfiles

JASMine Physical Storage Organization

- Store
  - A logical entity made up of libraries, servers, data movers, and a database.
- Library
  - A set of volumes and drives.
- Drive
  - A media reader or writer.
- Volume
  - A unit of storage media.

JASMine Services

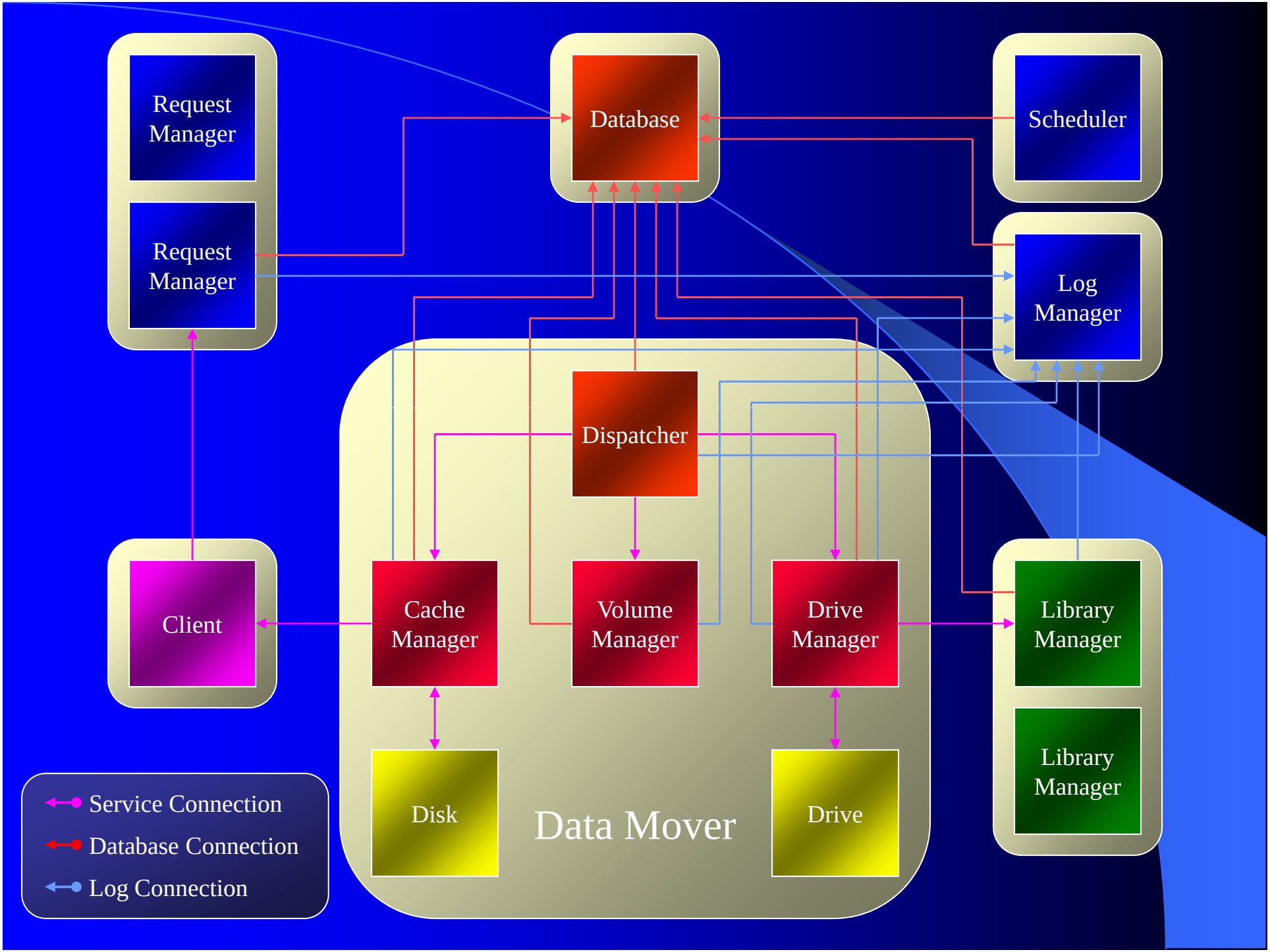
- Request Manager
  - Handles user requests and queries.
- Scheduler
  - Prioritizes user requests for tape access.
    - *priority = share / (.01 + (num_a * ACTIVE_WEIGHT) + (num_c * COMPLETED_WEIGHT))*
- Log Manager
  - Writes out log and error files and databases.
  - Sends out notices for failures.
- Library Manager
  - Mount and dismounts tapes as well as other library related tasks.

JASMine Services 2

- Data Mover
  - Dispatcher
    - Keeps track of available local resources and starts requests the local system can work on.
  - Cache Manager
    - Manages a disk or disks for pre-staging data to and from tape.
    - Sends and receives data to and from clients.
  - Volume Manager
    - Manages tapes for availability.
  - Drive Manager
    - Manages tape drives for usage.

User Access

- Jls
  - Get metadata for one or more files
- Jtstat
  - Status of the request queue
- Jput
  - Put one or more files on tape
- Jget
  - Get one or more files from tape
- Jcache
  - Copies one or more files from tape to cache


Request Manager
Request Manager
Database
Scheduler
Log Manager
Dispatcher
Client
Cache Manager
Volume Manager
Drive Manager
Library Manager
Library Manager
Disk
Data Mover
Drive
Service Connection
Database Connection
Log Connection

Why we needed a Disk Cache

- Tape Drives
  - Inefficient utilization.
- Batch Farm
  - Wasted CPU cycles.
  - No pre-staging of files to disk.
  - No post-staging of result files.
- Users
  - Requests were pending for days waiting for an available tape drive.

User Access

- NFS
  - Directory of links points the way.
  - Mounted read only by the farm and ifarm.
  - Users can mount read only on their desktop.
- Jcache
  - Java client.
  - Checks to see if files are on cache disks.
  - Will get/put files from/to cache disks.
    - Users can currently only get files.

Disk Cache Management

- Disk Pools are divided into groups
  - Tape staging.
  - Experiments.
  - Pre-staging for the batch farm.
- Management policy set per group
  - Cache – LRU files removed as needed.
  - Stage – Reference counting.
  - Explicit – manual addition and deletion.

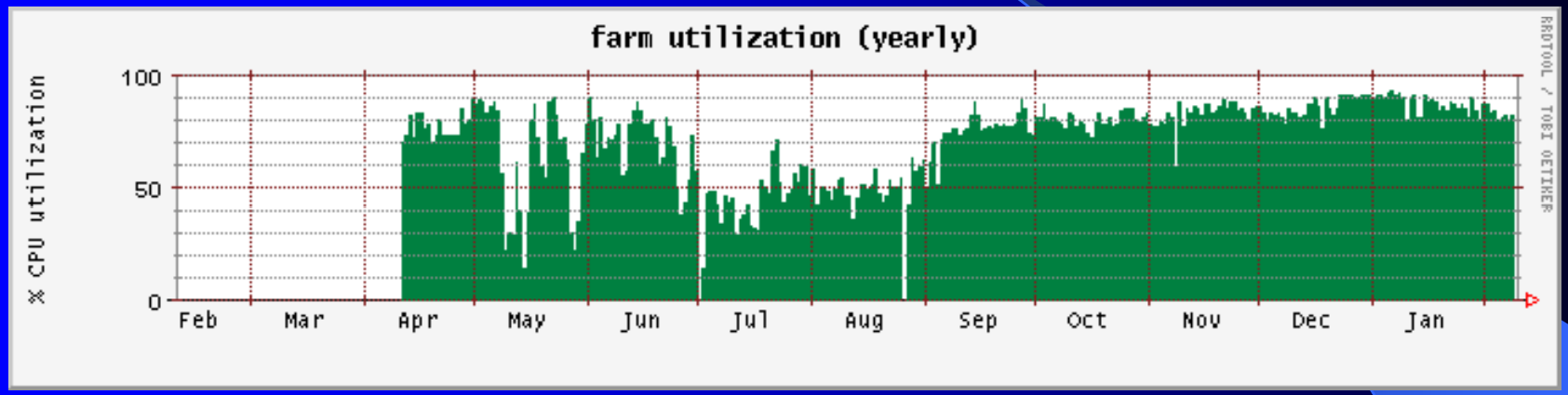
Benefits

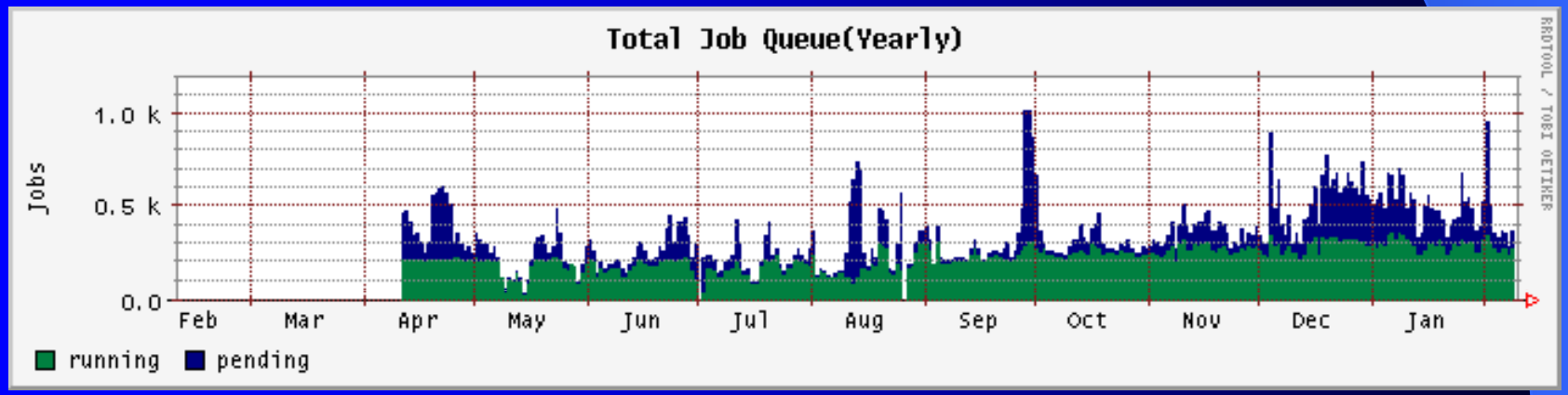
- To tape
  - Better utilize tape drive bandwidth.
  - User issued commands return sooner.
  - Allows tape software to better sort write requests.
  - Allows tape software to do retries without the client.
- From tape
  - Better utilize tape drive bandwidth.
  - Requests for cached files come from disk and thus reduce tape drive utilization.

Benefits cont.

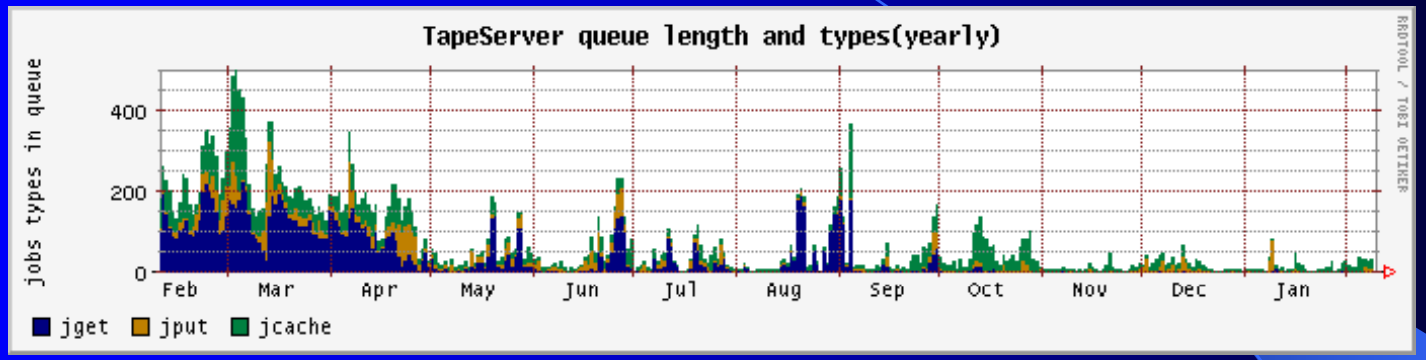
- Batch Farm
  - Pre-staging data files.
  - No wasted CPU cycles.
- Users
  - Few, if any, complaints about response time.

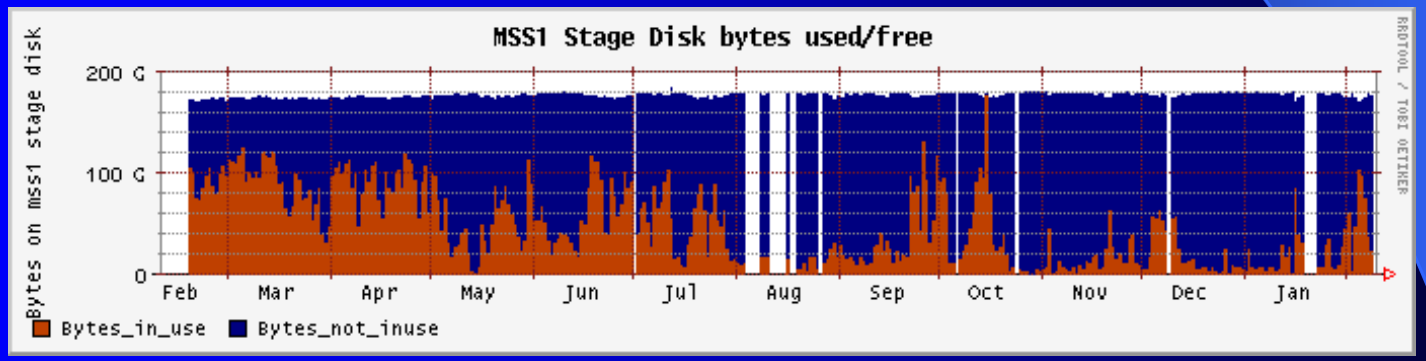
Benefits cont.

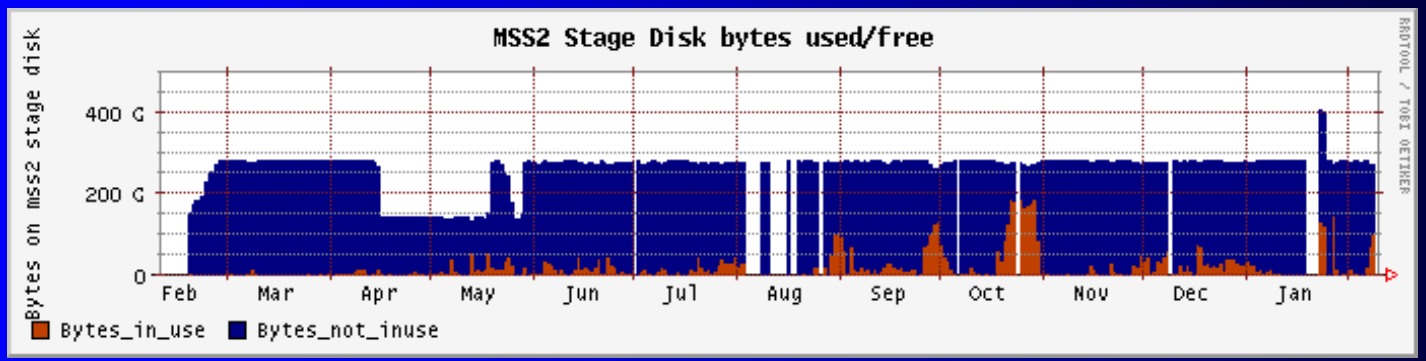




Benefits cont.







Protocol for file moving

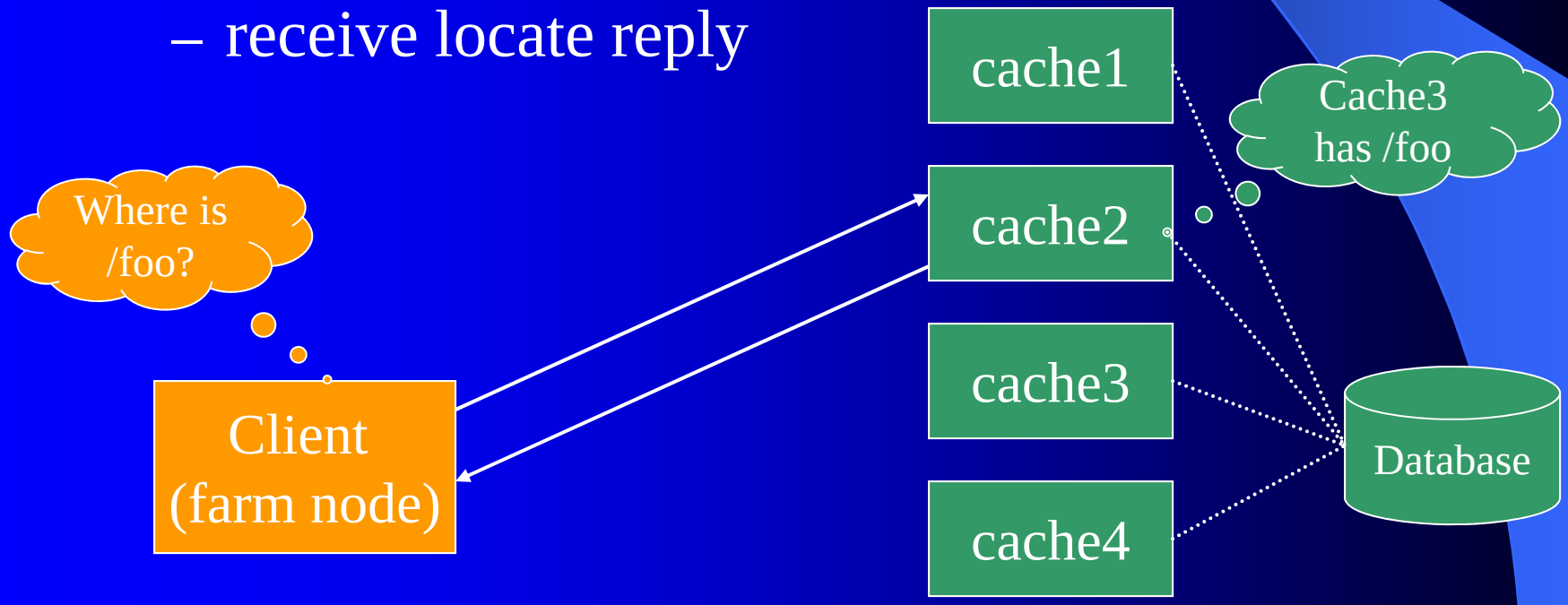
- Simple extensible protocol for file copies
- Messages are java serialized object
- Protocol is synchronous – all calls block
- Asynchrony by threading
- Fall back to raw data transfer for speed
- More fair than NFS
- Session may make many connections

Protocol for file moving

- Cache server extends the basic protocol
  - Add database hooks for cache
  - Add hooks for cache policies
  - Additional message type were added

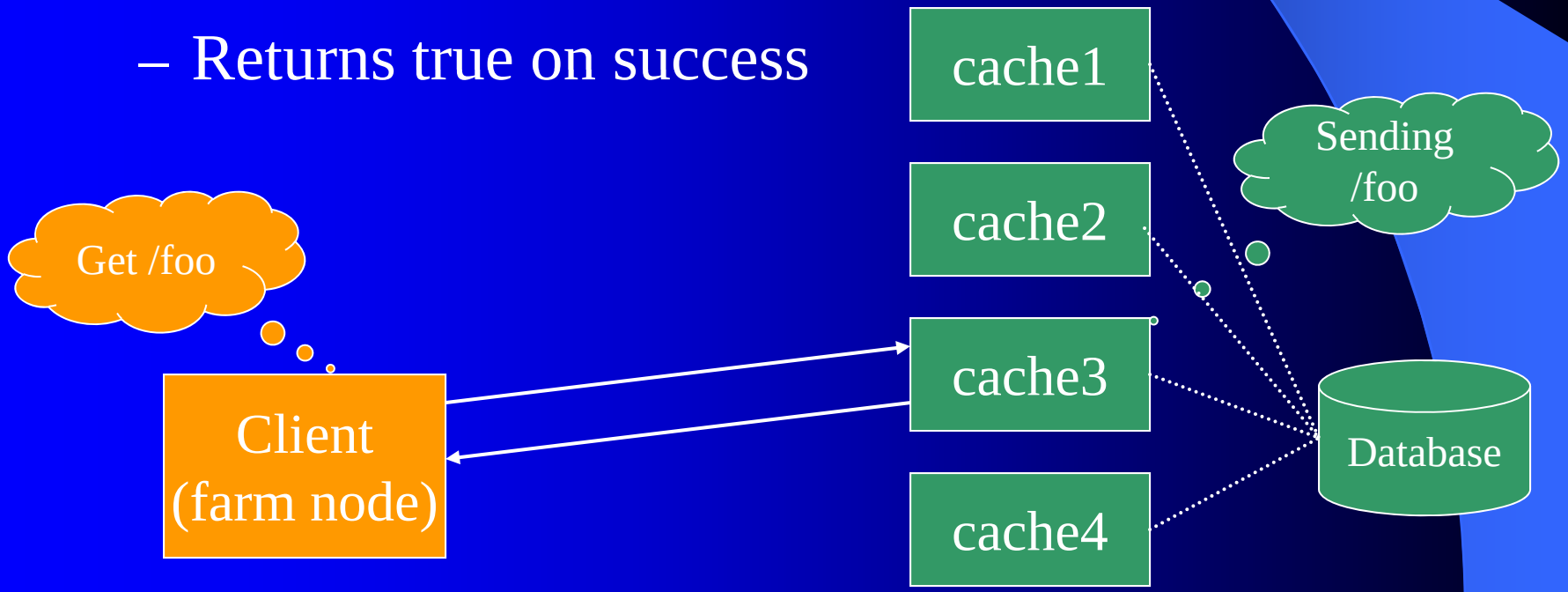
Example: Get from cache using our Protocol (1)

- cacheClient.getFile("/foo", "halla");
  - send locate request to any server
  - receive locate reply



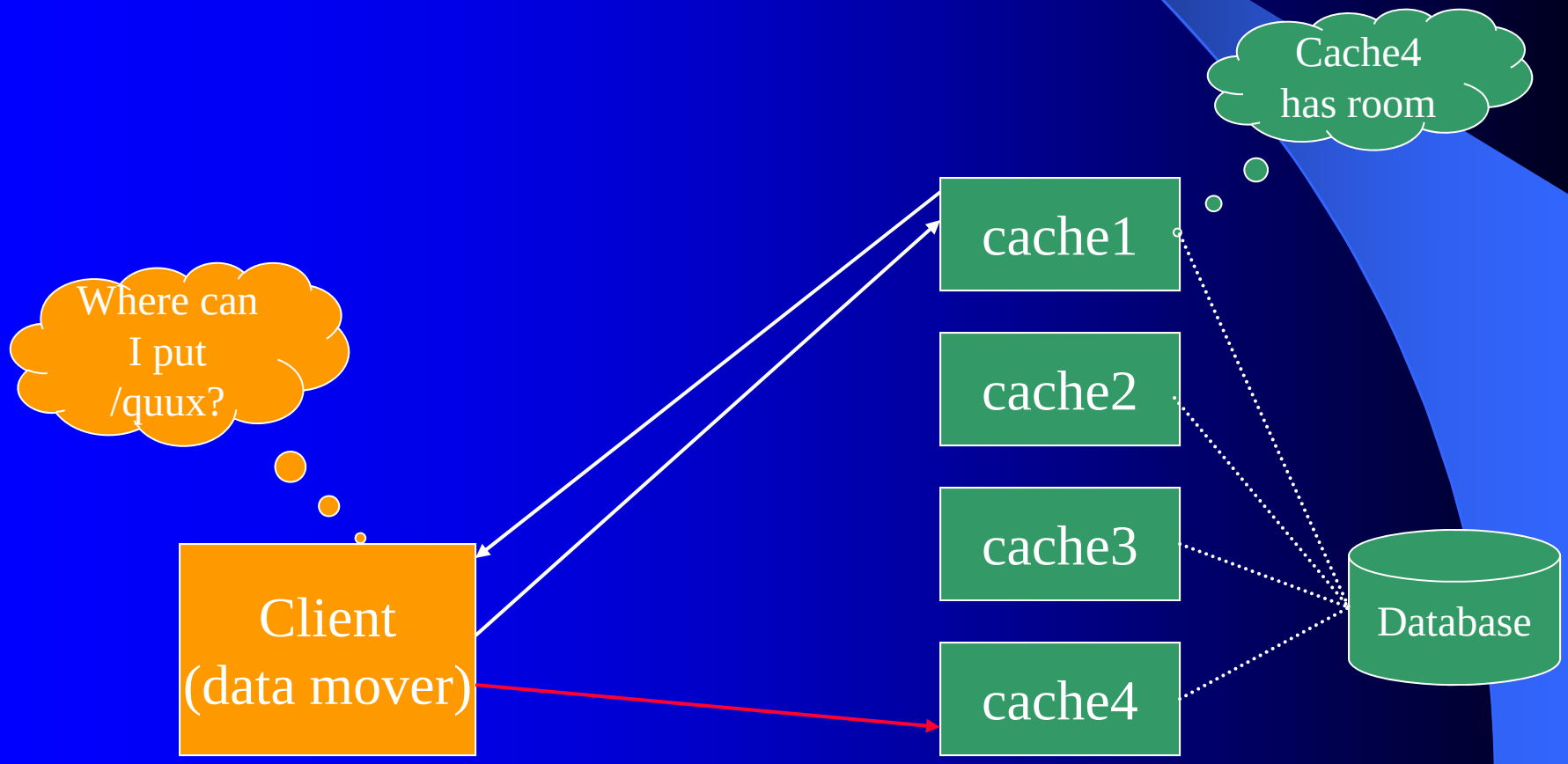
Example: Get from cache using our Protocol (2)

- cacheClient.getFile(“/foo”, “halla”);
  - contact appropriate server
  - initiate direct xfer
  - Returns true on success



Example: simple put to cache using our Protocol

- putFile(“/quux”,”halla”,123456789);



Fault Tolerance

- Dead machines do not stop the system
  - Data Movers work independently
  - Cache Servers will only impact NFS clients
- Exception handling for
  - Receive timeouts
  - Refused connections
  - Broken connections
  - Complete garbage on connections

Authorization and Authentication

- Shared secret for each file transfer session
  - Session authorization by policy objects
  - Example: receive 5 files from user@bar
- Plug-in authenticators
  - Establish shared secret between client and server
  - No clear text passwords

Bulk Data Transfers

- Model supports parallel transfers
  - Many files at once, but not bbftp style
  - For bulk data transfer over WANs
- Web-based class loader– zero pain updates
- Firewall issues
  - Client initiates all connections

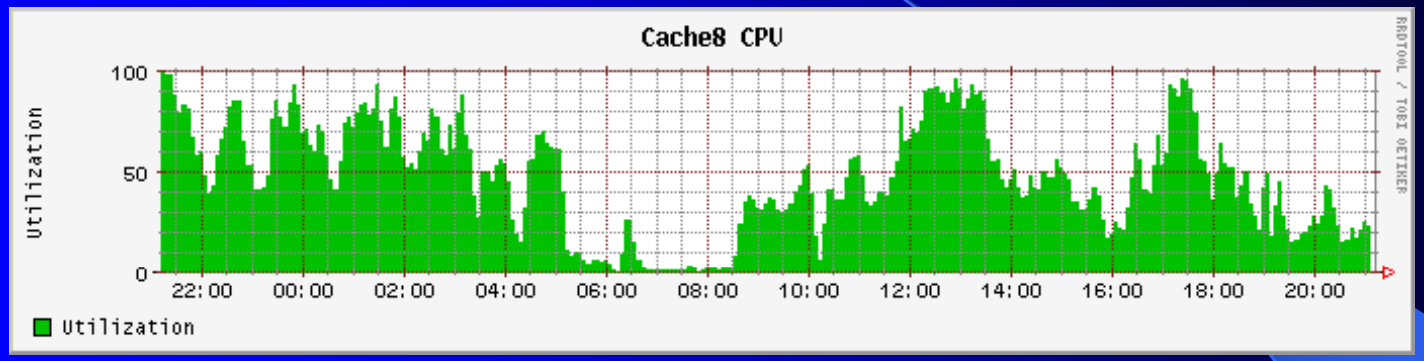
Architecture: Hardware

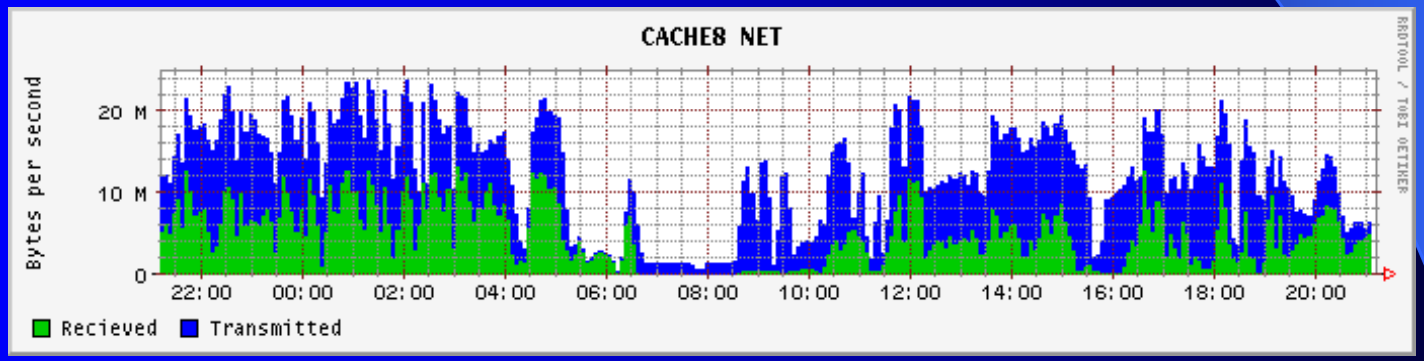
- SCSI Disk Servers
  - Dual Pentium III 650MHz CPUs
  - 512 Mbytes 100MHz SDRAM ECC
  - ASUS P2B-D Motherboard
  - NetGear GA620 Gigabit Ethernet PCI NIC
  - Mylex eXtremeRAID 1100, 32 MBytes cache
  - Seagate ST150176LW (Qty. 8) - 50 GBytes Ultra2 SCSI in Hot Swap Disk Carriers
  - CalPC 8U Rack Mount Case with Redundant 400W Power Supplies

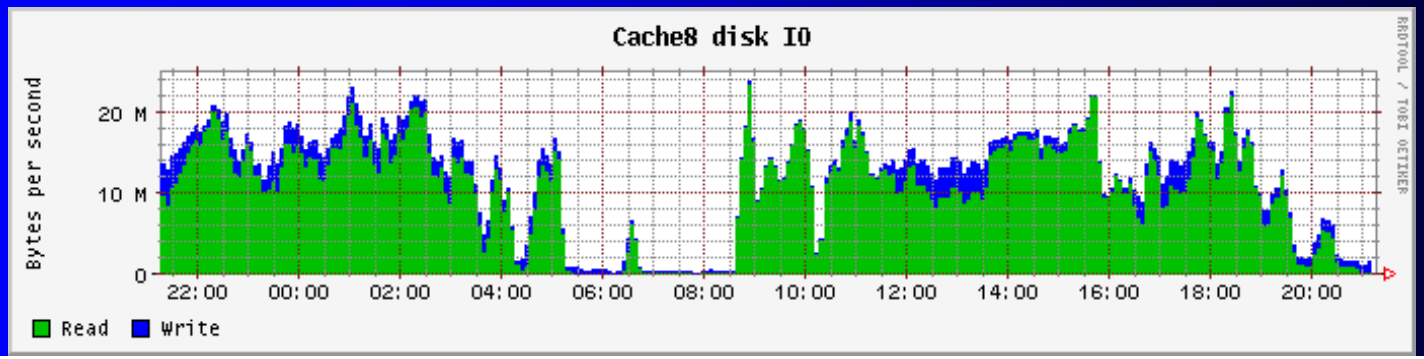
Hardware cont.

- IDE Disk Servers
  - Dual Pentium III 933MHz CPUs
  - 512 Mbytes 133MHz SDRAM ECC
  - Intel STL2 or ASUS CUR-DLS Motherboard
  - NetGear GA620 or Intel PRO/1000 T Server Gigabit Ethernet PCI NIC
  - 3ware Escalade 6800
  - IBM DTLA-307075 (Qty. 12) - 75 GBytes Ultra ATA/100 in Hot Swap Disk Carriers
  - CalPC 8U Rack Mount Case with Redundant 400W Power Supplies

Performance







Future Work

- Web interfaces for users
- Convert logging to syslog
- Replace RMI and other network calls with an XML based API
- Gateways to provide access to data via standard file copy protocols (ftp)
- Better Authentication
- Particle Physics Data Grid

Conclusions

- Java performs just as well as C for data movement
- Linux and PC hardware have a place here
- Distributed Data Movers is a must
- Disk Caches help overall performance
- These systems are fun to build