Managed Data Storage and Data Access Services for Data Grids

Jon Bakken FNAL
Ian Fisk FNAL
Patrick Fuhrmann DESY
Tigran Mkrtchyan DESY
Timur Perelmutov FNAL
Don Petravick FNAL

Michael Ernst
FNAL (until 03/2004) and DESY

Data Grid Challenge …

… as defined by the GriPhyN Project

"Global scientific communities, served by networks with bandwidths varying by orders of magnitude, need to perform computationally demanding analyses of geographically distributed datasets that will grow by at least 3 orders of magnitude over the next decade, from the 100 Terabyte to the 100 Petabyte scale."

Provide a new degree of transparency in how data is handled and processed

Characteristics of HEP Experiments

- Data is acquired at a small number of facilities
  Data is accessed and processed at many locations
- The processing of data and data transfers can be costly
- The scientific community needs to access both raw data as well as processed data in an efficient and well managed way on a national and international scale

Data Intensive Challenges Include

- Harness potentially large number of data, storage, network resources located in distinct administrative domains
- Respect local and global policies governing usage Schedule resources efficiently, again subject to local and global constraints
- Achieve high performance, with respect to both speed and reliability
- Discover “best” replicas

The Data Grid

Three major components:

1. Storage Resource Management
   - Data is stored on Disk Pool Servers or Mass Storage Systems
   - Storage resource Management needs to take into account
     - Transparent access to files (migration from/to disk pool)
     - File Pinning
     - Space Reservation
     - File Status Notification
     - Life Time Management

   - Storage Resource Manager (SRM) takes care of all these details
     - SRM is a Grid Service that takes care of local storage interaction and provides a Grid Interface to off-site resources

The Data Grid

Three major components:

1. Storage Resource Management (cont'd)
   - Support for local policy
     - Each Storage Resource can be managed independently
       - Internal priorities are not sacrificed by data movements between Grid Agents
   - Disk and Tape resources are presented as a single element
   - Temporary Locking / Pinning
     - Files can be read from disk caches rather than from tape
   - Reservation on demand and advance reservation
     - Space can be reserved for registering a new file
     - Plan the storage system usage
   - File Status and Estimates for Planning
     - Provides Information on File Status
     - Provides Information on Space Availability / Usage

The Data Grid

Three major components:

1. Storage Resource Management (cont'd)
   - SRM provides a consistent interface to Mass Storage regardless of where data is stored (Secondary and/or Tertiary Storage)
   - Advantages
     - Adds resiliency to low level file transfer services (i.e. FTP)
       - Restarts transfer if hung
       - Checksums
     - Traffic Shaping (to avoid oversubscription of servers and networks)
     - Credential Delegation in 3rd party transfer
     - … over POSIX: File Pinning, Caching, Reservation
   - Current Limitations
     - Standard does not include access to objects in a file
       - POSIX file system semantics (e.g. seek, read, write) are not supported
       - Need to use additional file I/O lib to access files in the storage system (details on GFAL by Jean-Philippe this session at 3:40 PM)
   - More on SRM and SRM based Grid SE
     - Patrick Fuhrmann on Wed. at 4:40 PM in Computer Fabrics track
     - Timur Perelmutov on Wed. at 5:10 PM in Computer Fabrics track

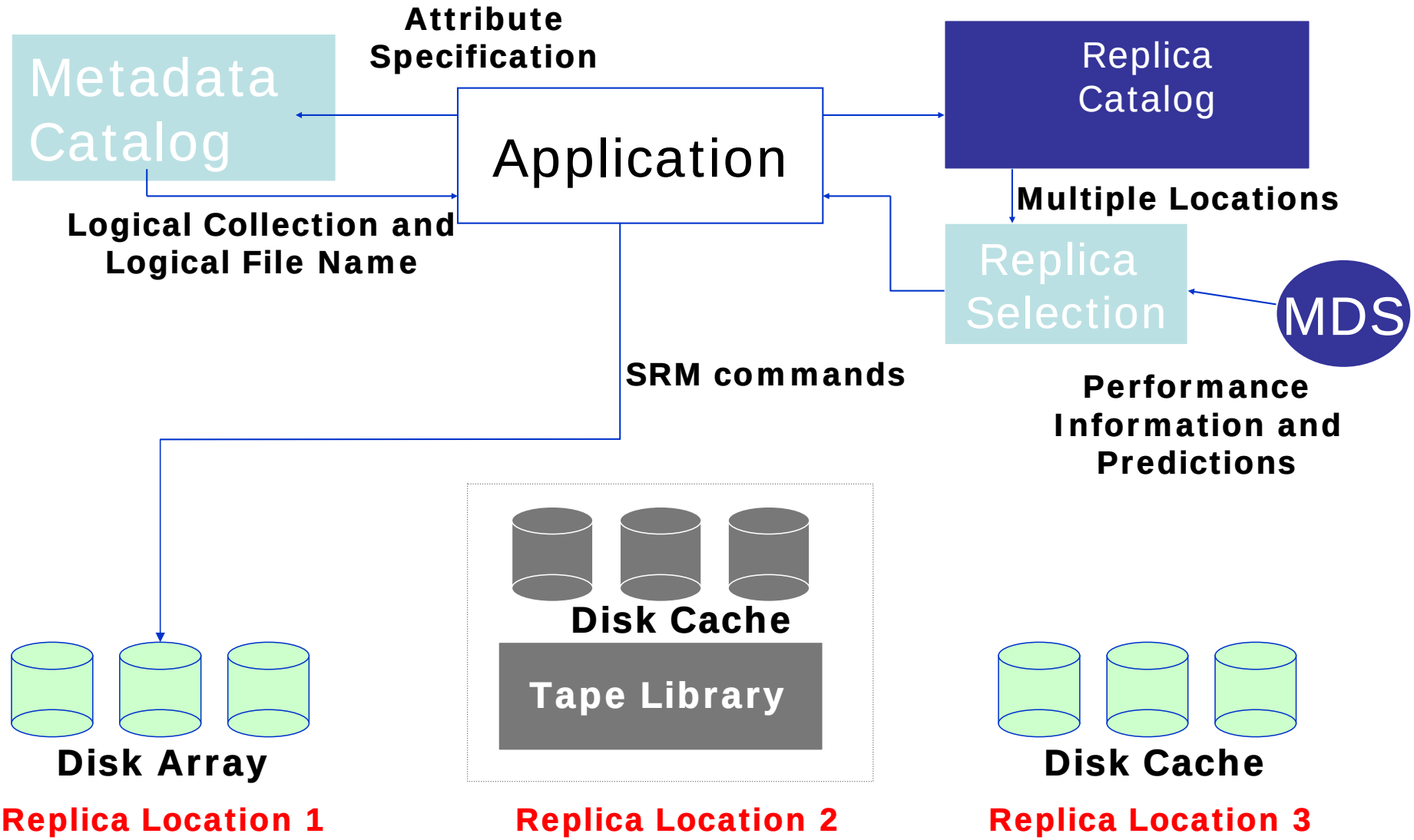
The Data Grid

Three major components:

2. Data Transport and Access, GridFtp
   - Built on top of ftp
   - Integrated with the Grid Security Infrastructure (GSI)
   - Allows for third-party control and data transfer
   - Parallel data transfer (via multiple TCP streams)
   - Striped data transfer support for data striped or interleaved across multiple servers
   - Partial file transfer
   - Restartable data transfer

3. Replica Management Service
   - Simple scheme for managing
     - multiple copies of files
     - collections of files

A Model Architecture for Data Grids



Facilities and Grid Users need managed Data Services

- The facility provider should not have to rely upon the application to clean and vacate storage space
- Current architecture has bottlenecks associated with IO to the clusters
- Difficult for facility providers to enforce and publish storage usage policies using scripts and information providers.
- Difficult for facilities to satisfy obligations to VOs without storage management and auditing
- Difficult for users to run reliably if they cannot ensure there is a place to write out the results
  - Even more important as applications with large input requirements are attempted

Storage Elements on Facilities

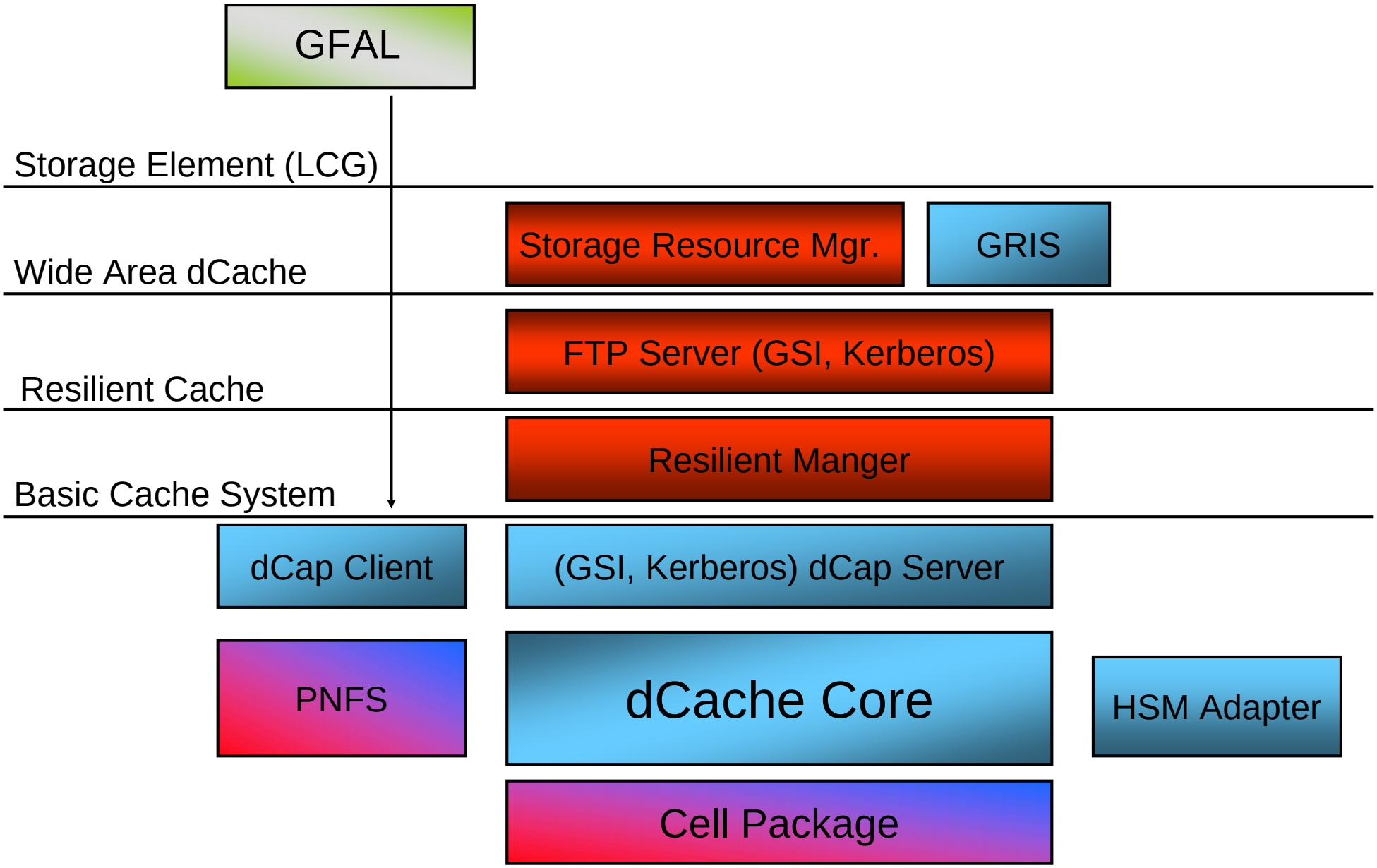
- The basic management functionality is needed on the cluster regardless of
- how much storage is there
  - A large NFS mounted disk area still needs to be cleaned up and and application needs to be able to notify the facility how long it needs to have files stored, etc.
  - Techniques for transient storage management need to be developed
    - Not the purpose of this talk

- SRM + dCache provides most of the functionality described earlier
  - Installation is a little intrusive
  - This is the equivalent of the processing queue and makes equivalent requirements
  - This storage element has some very advanced features

SRM/dCache – A brief Introduction

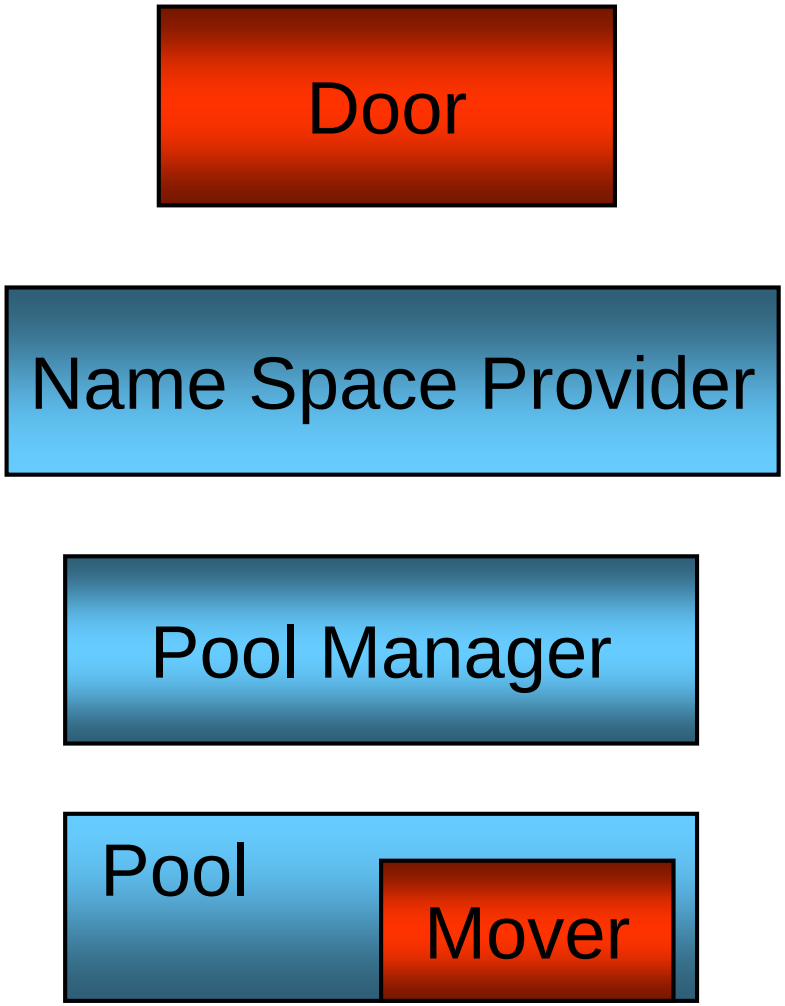
- SRM/dCache
  - Provides the storage
    - Physical disks or arrays are combined into a common filesystem
  - POSIX compliant interface
    - Unix LD_PRELOAD library or access library compiled into the application
  - Handles load balancing and system failure and recovery
    - Application waits patiently while file staged from MSS (if applicable)
  - Provides a common interface to physical storage systems
    - Virtualizes interfaces and hides detailed implementation
      - Allows migration of technology
  - Provides the functionality for storage management
  - Supervises and manages transfers
  - Circumvents GridFTP scalability problem

dCache Functionality Layers



dCache Basic Design

Components involved in Data Storage and Data Access



- Provides specific end point for client connection
- Exists as long as client process is alive
- Client's proxy used within dCache

Interface to a file system name space

- Maps dCache name space operations to filesystem operations
- Stores extended file metadata

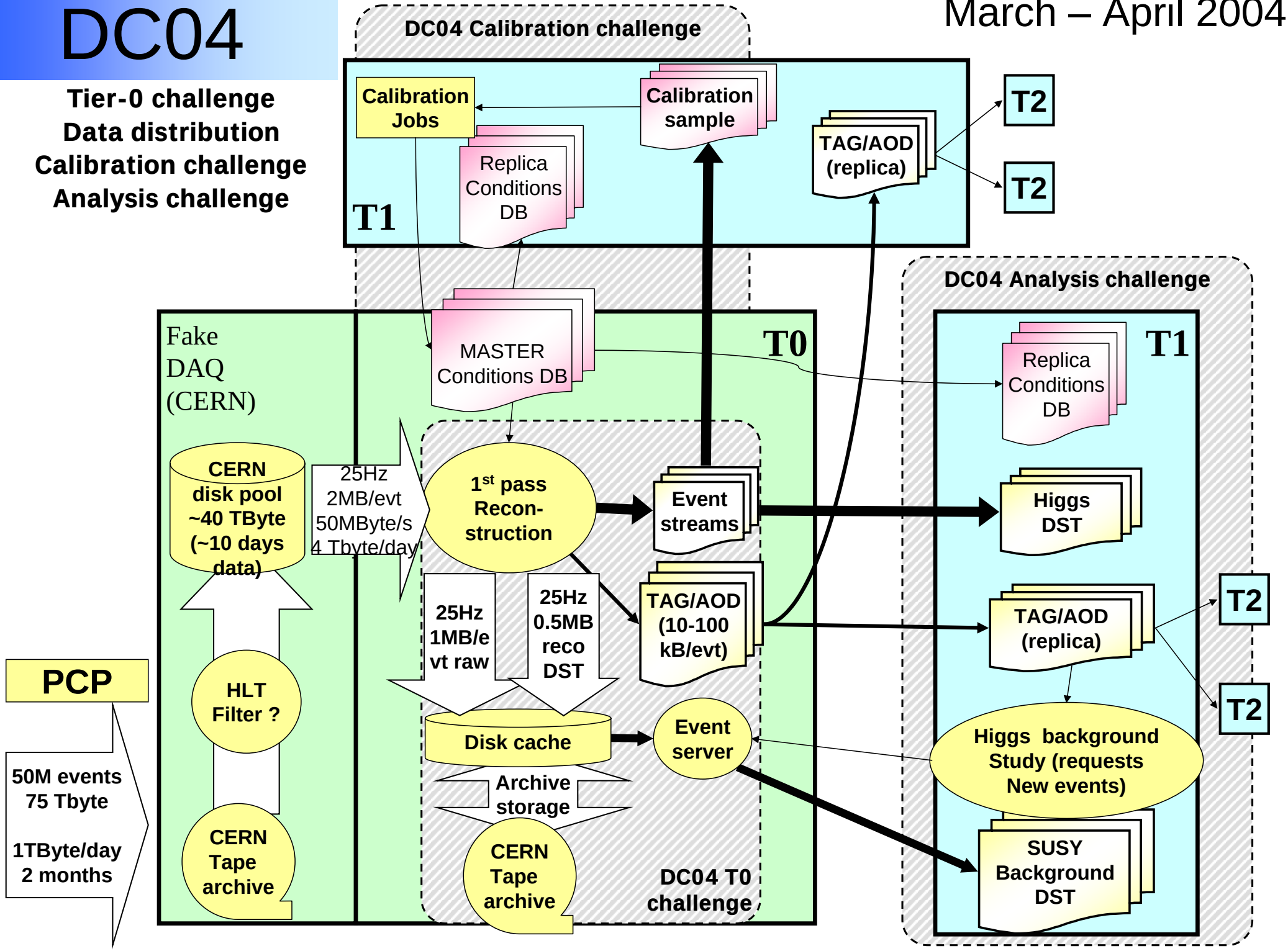
Performs pool selection

- Data repository handler
- Launches requested data transfer protocols
- Data transfer handler
  (gsi)dCap, (Grid)FTP, http, HSM hooks

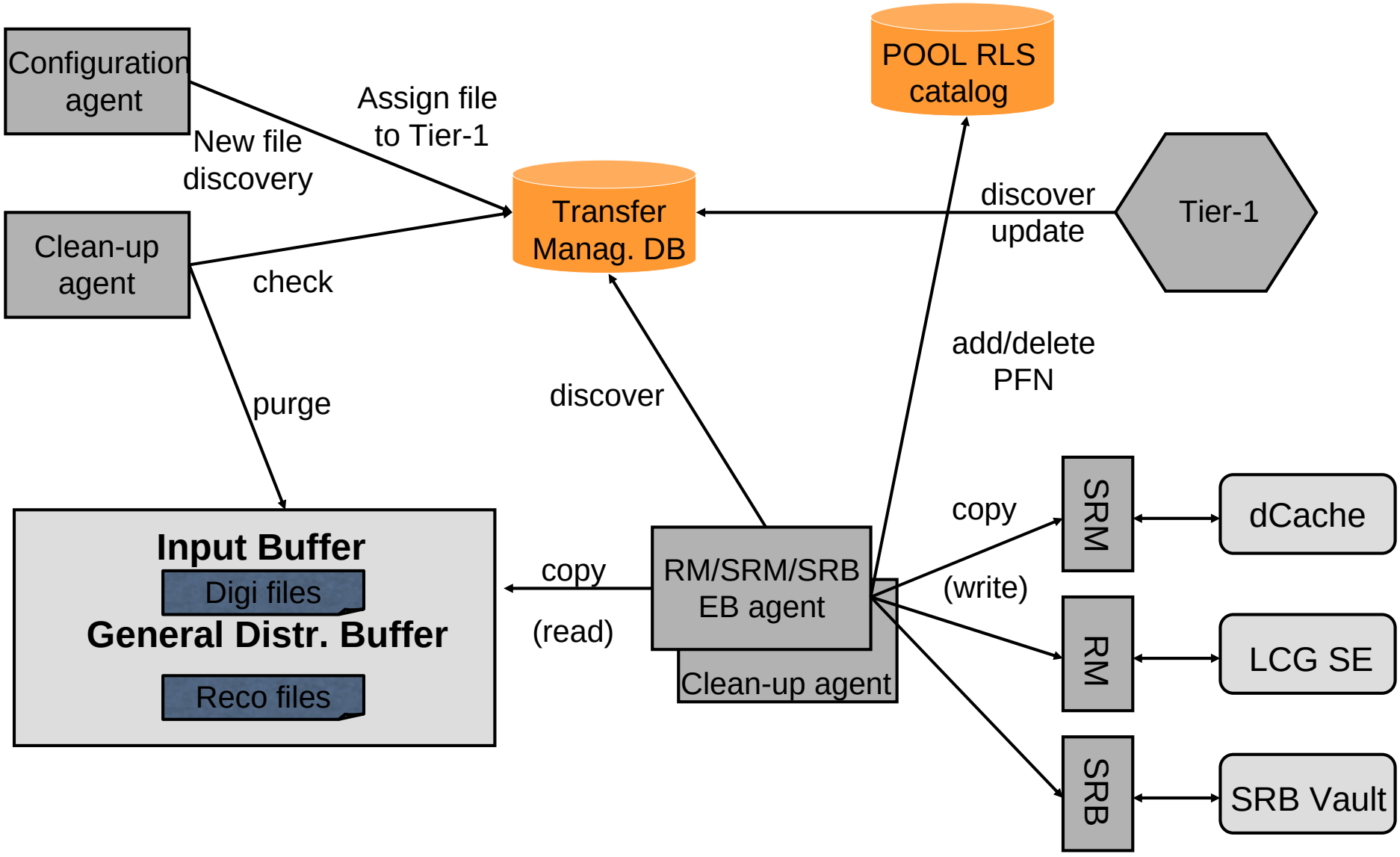
DC04

March – April 2004

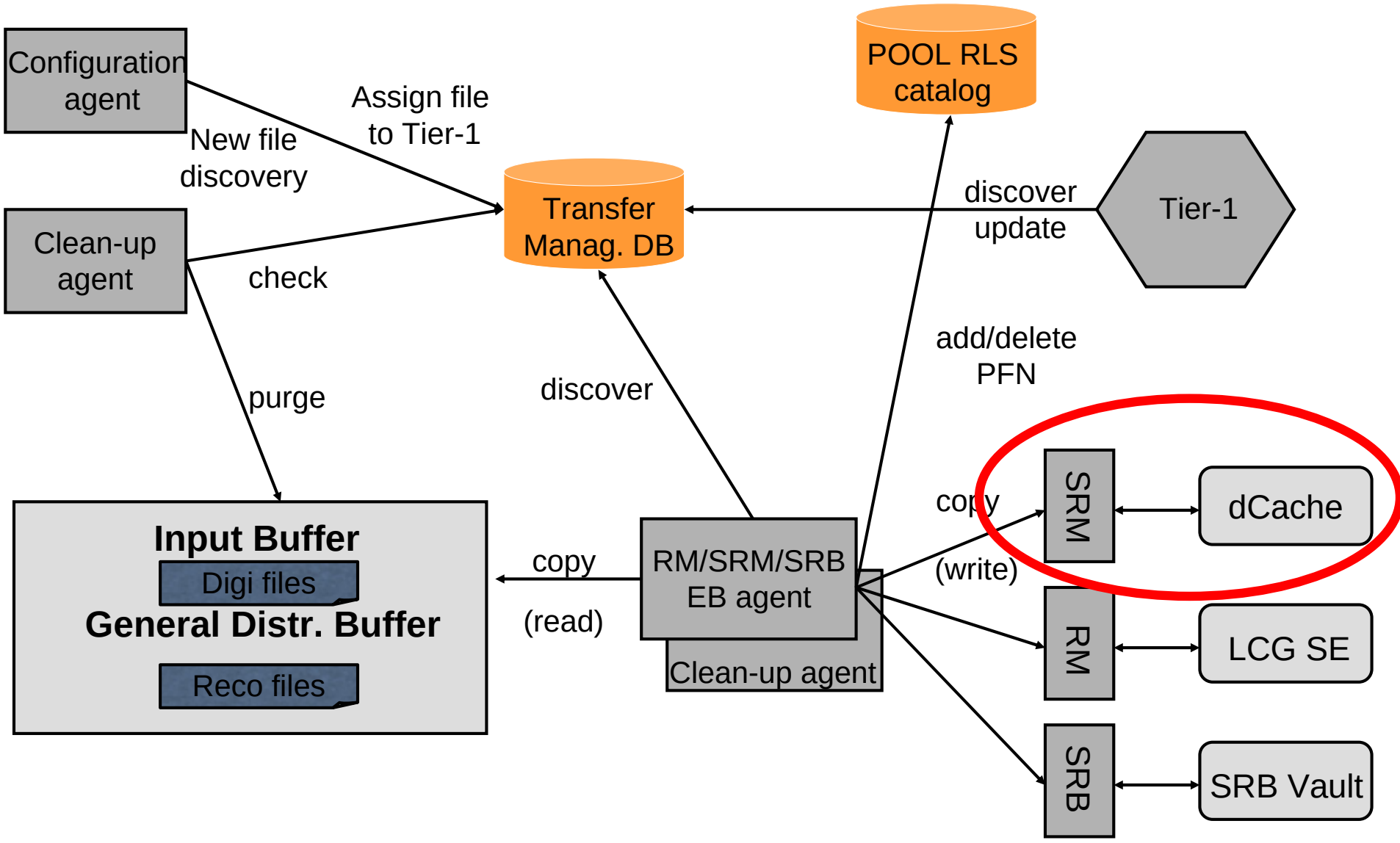
Tier-0 challenge
Data distribution
Calibration challenge
Analysis challenge



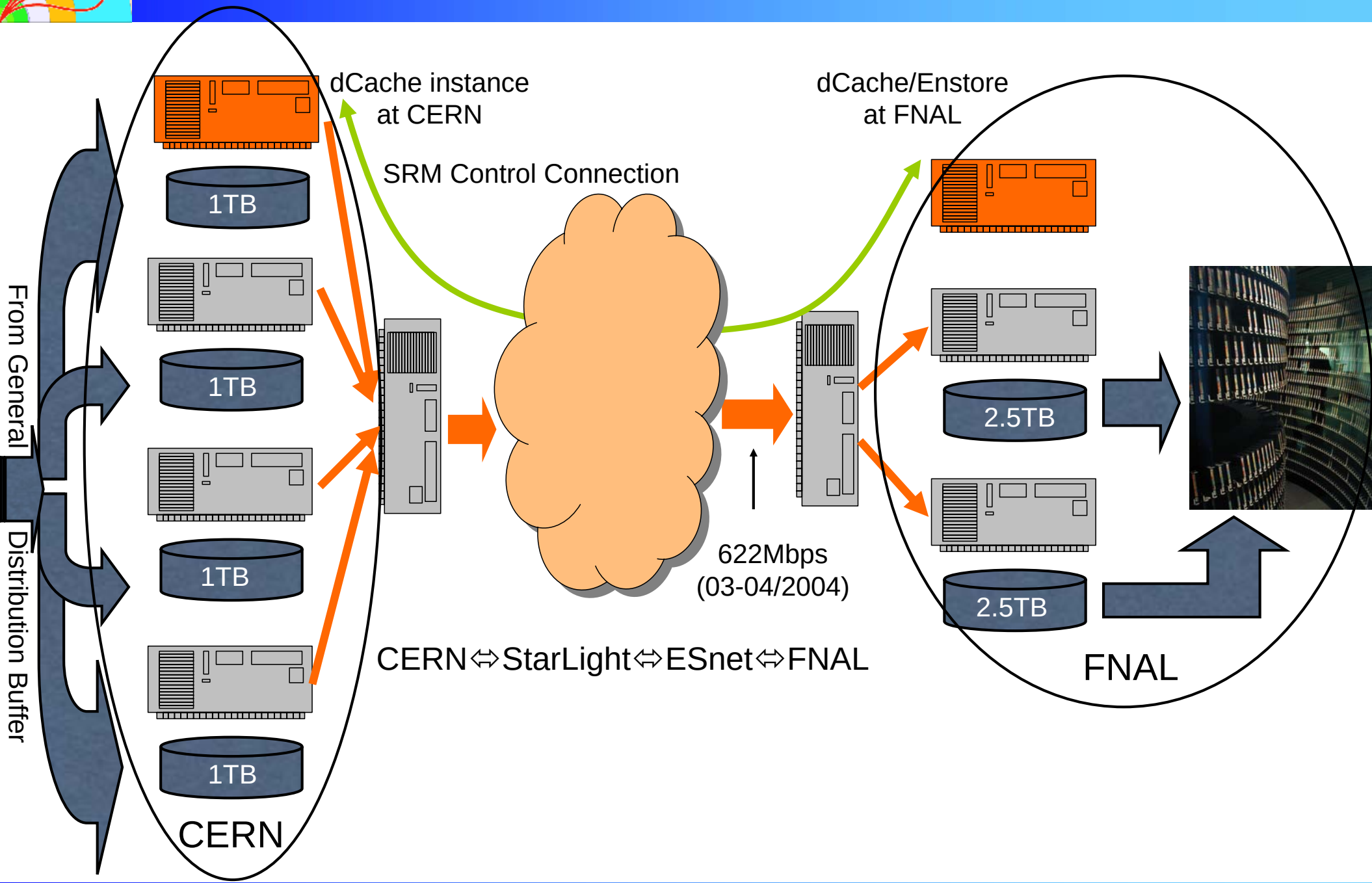
CMS DC04 Distribution Chain (CERN)



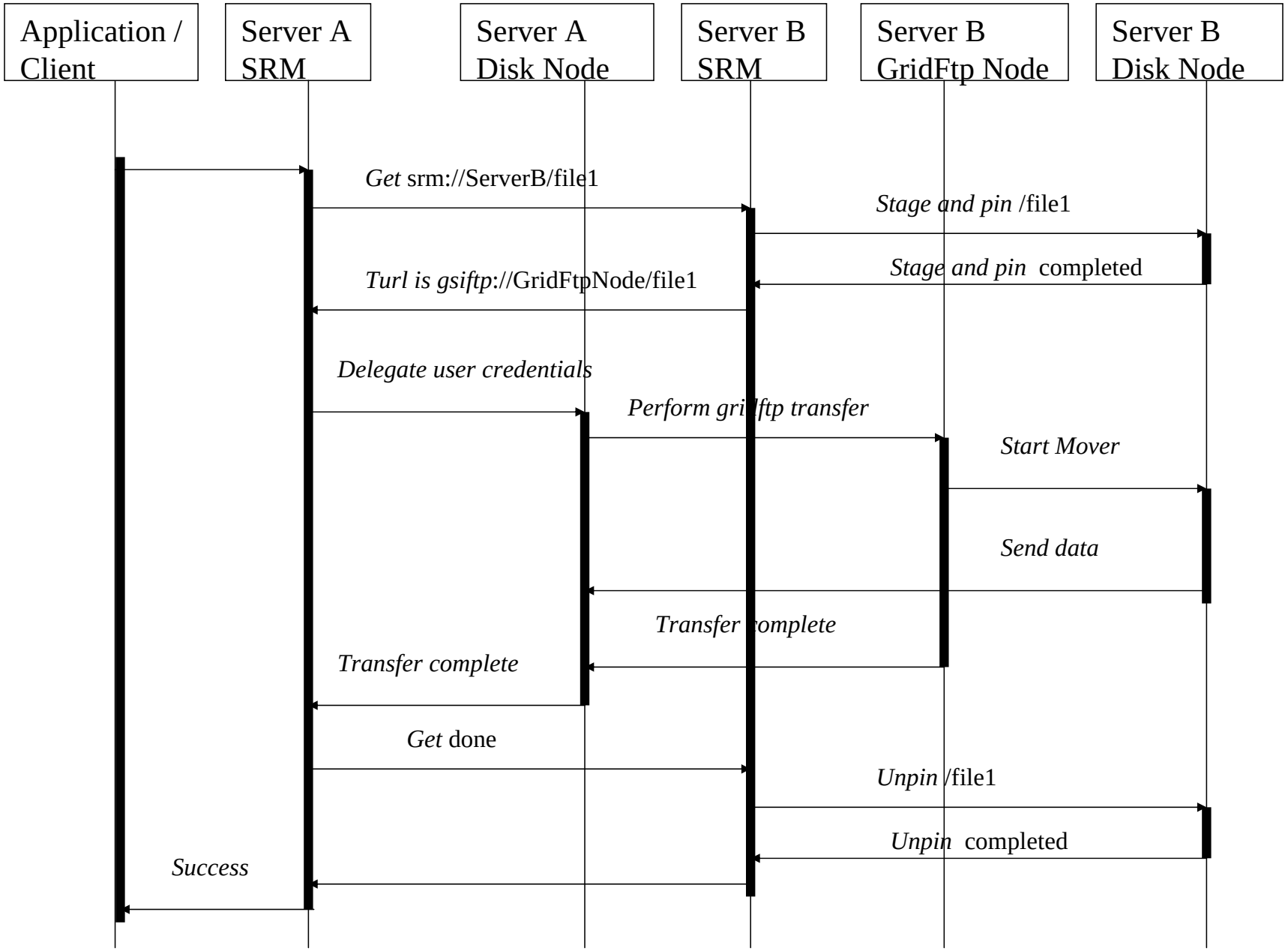
CMS DC04 Distribution Chain



CMS DC04 SRM Transfer Chain



The sequence diagram of the SRM Copy Function performing "Copy srm://ServerB/file1 srm://ServerA/file1"

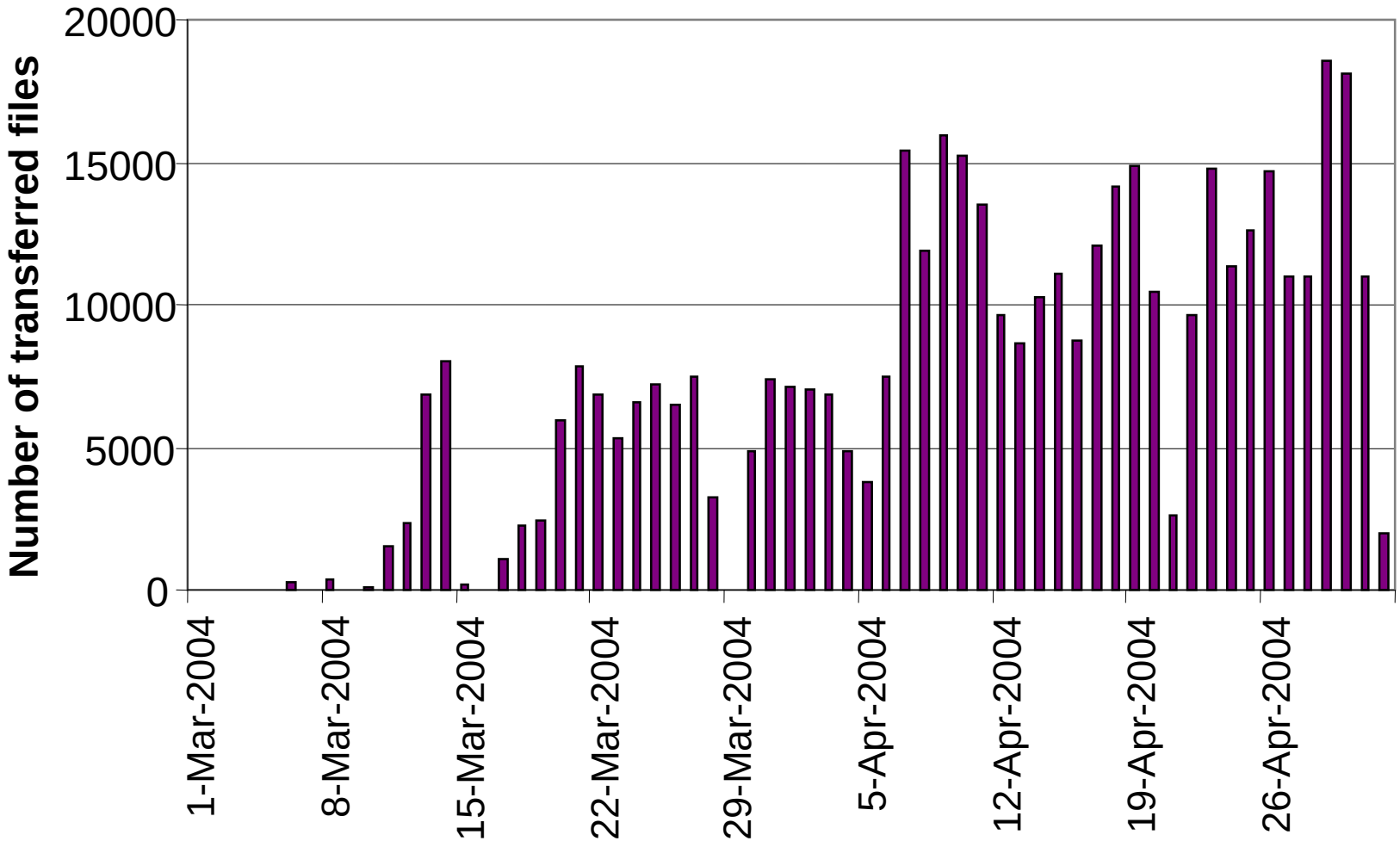


Summary on DC04 SRM transfer

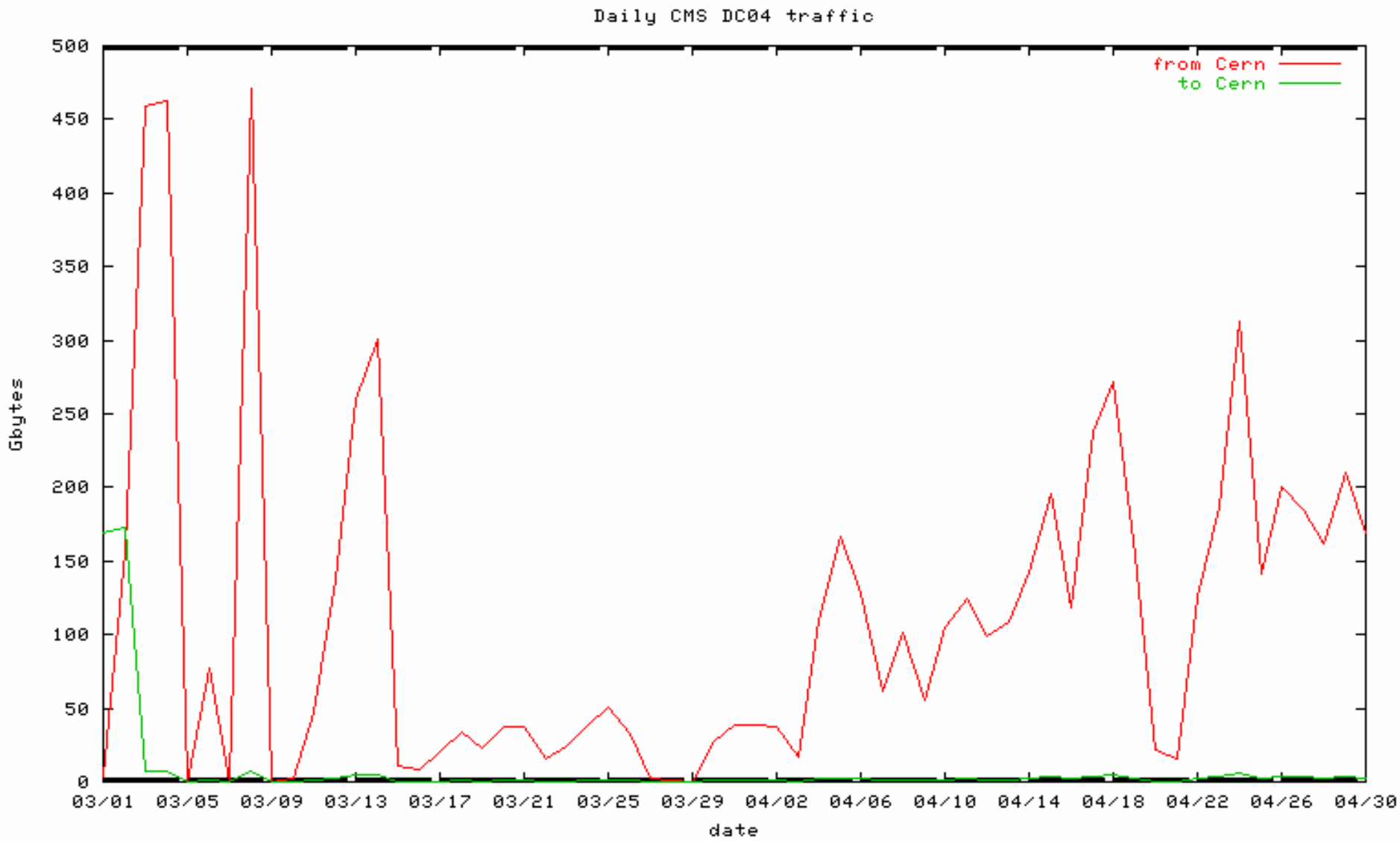
- Total data transferred to FNAL: 5.2TB (5293GB)
- Total number of files transferred: 440K
- Best transfer day in number of files: 18560
  - **Most of the files transferred in the first 12 hours, then idling---waiting for files to arrive at EB.**
- Best transfer day in size of data: 316254MB
- Average filesize was very small:

  ***min 20.8KB *max: 1607.8MB *mean: 13.2MB *median: 581.6KB**

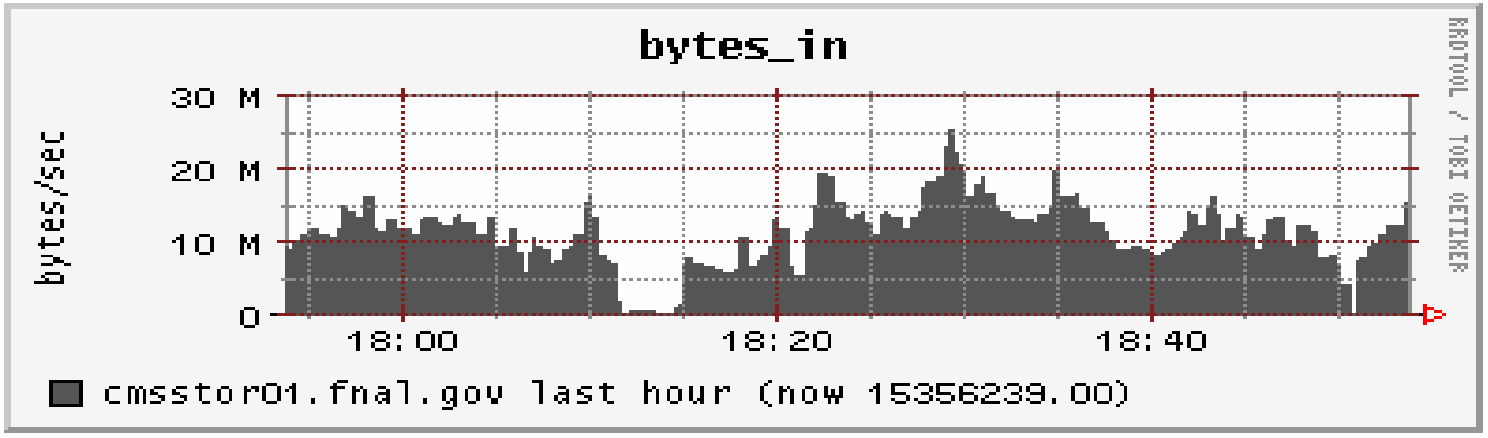
Number of transferred files in DC04 (CERN => FNAL)

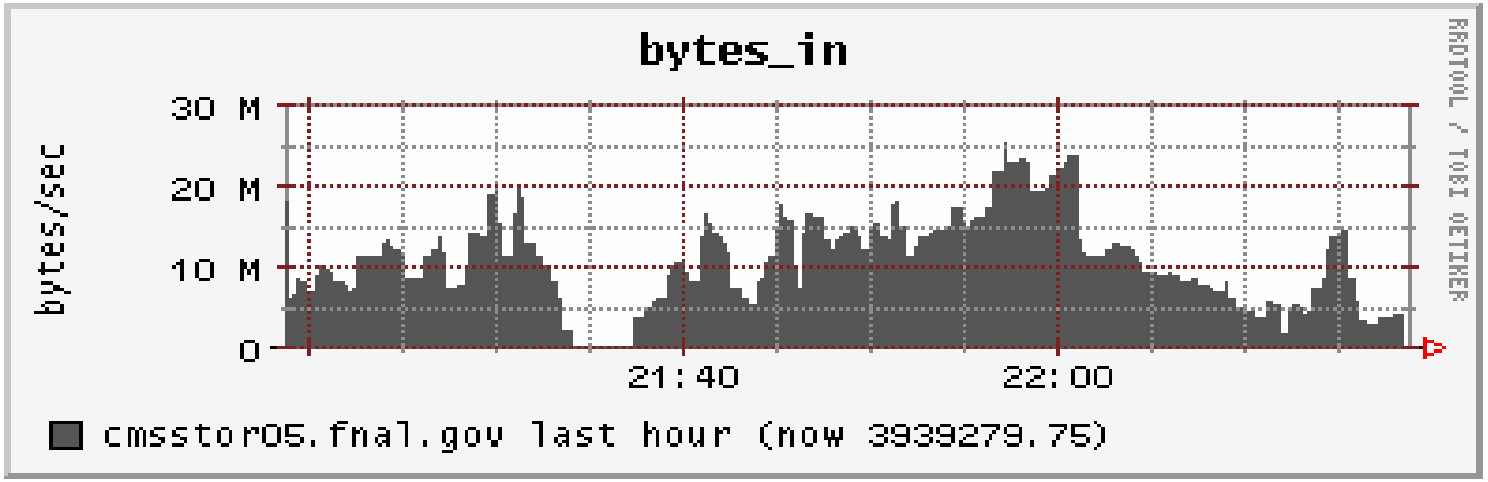


Daily data transferred to FNAL



dCache pool nodes network traffic





Experience

- We used multiple streams (GridFTP) with multiple files per SRM copy command to transfer files:
  - **15 srmcp (gets) in parallel and 30 files in one copy job for a total of 450 files per transfer;**
  - **This reduced the overhead of authentication and increased the parallel transfer performance;**
- SRM file transfer processes can survive network failure, hardware components failure without any problem
- Automatic file migration from disk buffer to tape

- We believe with the shown SRM/dCache setup 30K files/day and a sustained transfer rate of 20 – 30 MB/s is achievable

Some things to improve …

Srmcp batches: Transfer scheduler aborts all if single transfer fails (solved in latest version)

Client failure: Supposed to retry transfer in case of a pool failure, selecting a different pool (solved)

Space reservation: Prototype available for SC2003; needs to be integrated with SRM v1.x (planned for Q4/2004)

Information Provider: Need a tightly integrated information provider for optimization

Future Development

- HEP Jobs are data-intensive → important to take data location into account
- Need to integrate scheduling for large-scale data intensive problems in Grids
- Replication of data to reduce remote data access

Vision for Next Generation Grids

Design goal for current Grid development:

Single generic Grid infrastructure
providing simple and transparent access
to arbitrary resource types
supporting all kinds of applications

➔contains several challenges for Grid scheduling and (storage) resource management

Grid (Data) Scheduling

- Current approach:
  - Resource discovery and load-distribution to a remote resource
  - Usually batch job scheduling model on remote machine
- But actually required for Grid scheduling is:
  - Co-allocation and coordination
    of different resource allocations for a Grid job
  - Instantaneous ad-hoc allocation not always suitable
- This complex task involves:
  - Cooperation between different resource providers
  - Interaction with local resource management systems
  - Support for reservation and service level agreements
  - Orchestration of coordinated resources allocation

Example: Access Cost for HSM System

- Depends on
  - Current load of HSM system
  - Number of available tape drives
  - Performance characteristics of tape drives
  - Data location (cache, tape)
  - Data compression rate

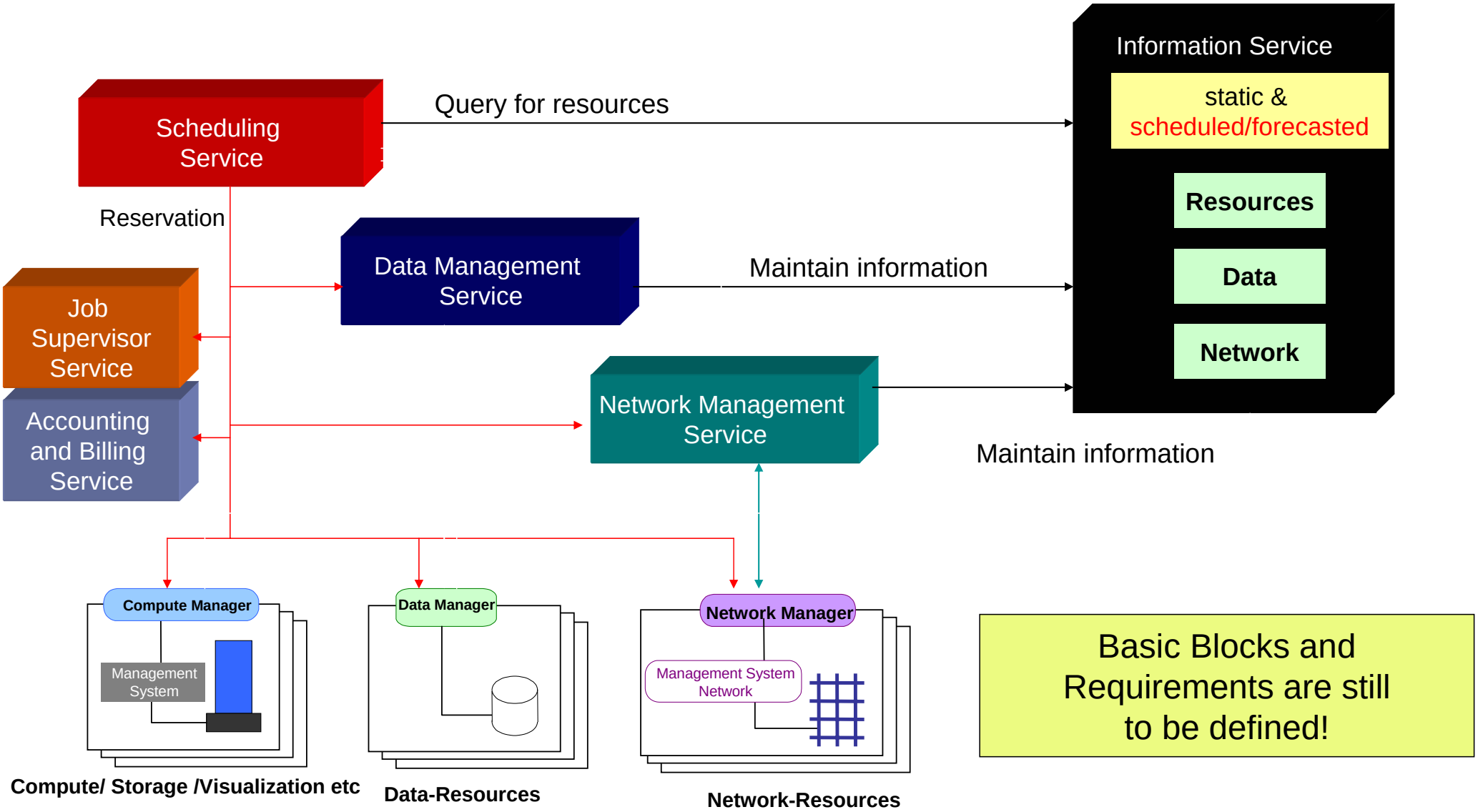
➢ Access_cost_storage = time_latency + time_transfer

time_latency = tw + tu + tm + tp + tt + td

time_transfer = size_file / transfer_rate_cache

Waiting for resoures
Unloading idle tape
Mounting tape
Positioning
Transfer tape => disk
Disk cache latency

Basic Grid Scheduling Architecture



Grid-specific Development Tasks

- Investigations, development and implementation of Algorithms required for decision making process
- “Intelligent” Scheduler
- Methods to pre-determine behavior of a given resource, i.e. a Mass Storage Management System by using statistical data from the past to allow for optimization of future decisions
- Current implementation requires the SE to act instantaneously on a request – Alternatives allowing to optimize resource utilization include
  - Provisioning (make data available at a given time)
  - Cost associated with making data available at a given time – defined cost metric could be used to select the least expensive SE
  - SE could provide information as to when would be the most optimal time to deliver the requested data

➢ In collaboration with Computer Scientists of Dortmund University and others within D-Grid (e-science program in Germany) initiative

Summary

- SRM/dCache based Grid enabled SE ready to serve HEP community
- Provide end to end, fault tolerance, run-time adaptation, multilevel policy support, reliable and efficient transfers

- Improve Information Systems and Grid schedulers to serve specific needs in Data Grids (Coallocation and Coordination)

- More Information
  - dCache http://www.dcache.org
  - SRM http://sdm.lbl.gov
  - EGEE http://www.eu-egee.org