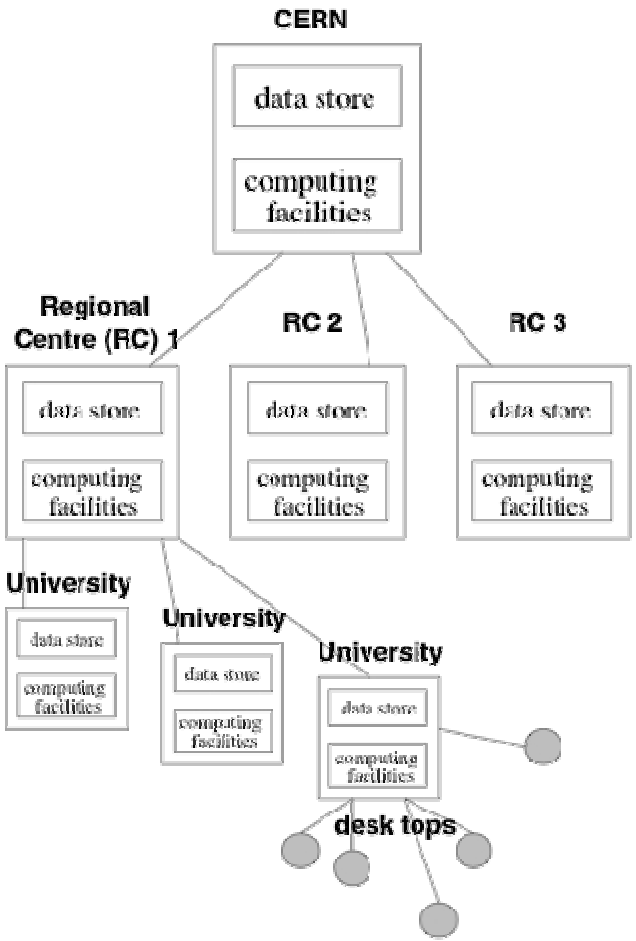
Data Management in an International Data Grid Project

Timur Chabuk
04/09/2007

Intro



- LHC opened in 2005
  - several Petabytes of data per year
  - data created at CERN
  - distributed to “Regional Centers” all over the world
- How to manage/store this much data?

“Research and Technological Development for an International Data Grid"

- Goals:
  - develop Research Network
  - demonstrate effectiveness through end-to-end applications
  - demonstrate ability to build from commodity components

- Data Management work package:
  - universal namespace
  - efficient data transfer between sites
  - synchronization of remote copies
  - wide-area data access/caching
  - interface to mass storage management systems

Related: Legacy

- AFS/NFS (distributed file systems)
  - interface for remote I/O, uniform name space
  - no multi-site replication, collective I/O

- Vesta and Galley
  - provides collective I/O
  - doesn't address wide area environment issues:
    - complex configurations, security, performance trade-offs

- Remote Execution Systems
  - location-independent execution of tasks scheduled to remote computers
  - no parallel I/O or access to parallel file systems

- Distributed Database Research
  - focused on synchronization of single transactions
  - not focused on moving large amounts of data

Related: Grid Computing

- Globus : Global Access to Secondary Storage
  - remote file I/O, local cache management, client-server model of file transfers
  - current work: replica management, optimized file transfers over wide area networks

- Legion
  - no explicit modules for data management issues
  - data management functionality via the backing store “vault” mechanism

Related: Grid Data

- Particle Physics Data Grid
  - develop basic infrastructure
  - high speed data transfers, transparent access
  - replica management, interfacing w/ different storage brokers
- GriPhyN: concept of “virtual data”
- SRB
  - uniform interface to different storage systems
  - access data via attributes (MCAT)
- China Clipper
  - high speed, integrated views of multiple data archives
  - resource discovery, monitoring
  - flexible management of access control / policy enforcement

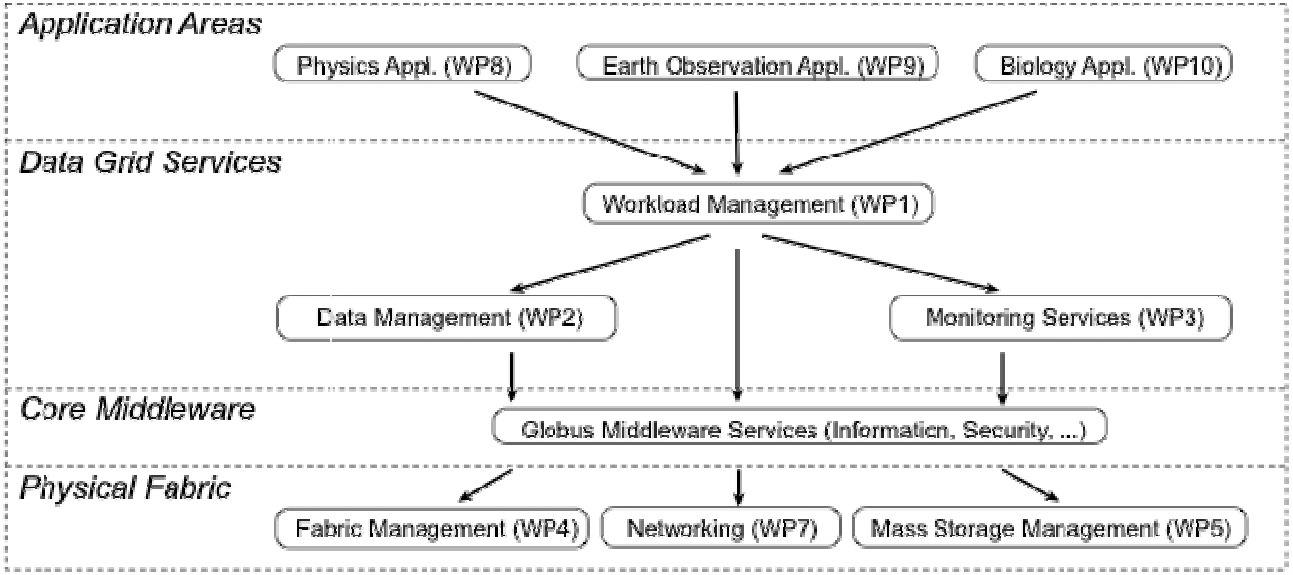
Use Cases

- High Energy Physics
  - 2000 distributed scientists analyze data generated from one source. dynamic distribution of data.
- Earth Observation
  - data collected from distributed sources, maintained in distributed sources
- Bioinformatics
  - large number of independent databases, integrated into one logical system

Common Aim:

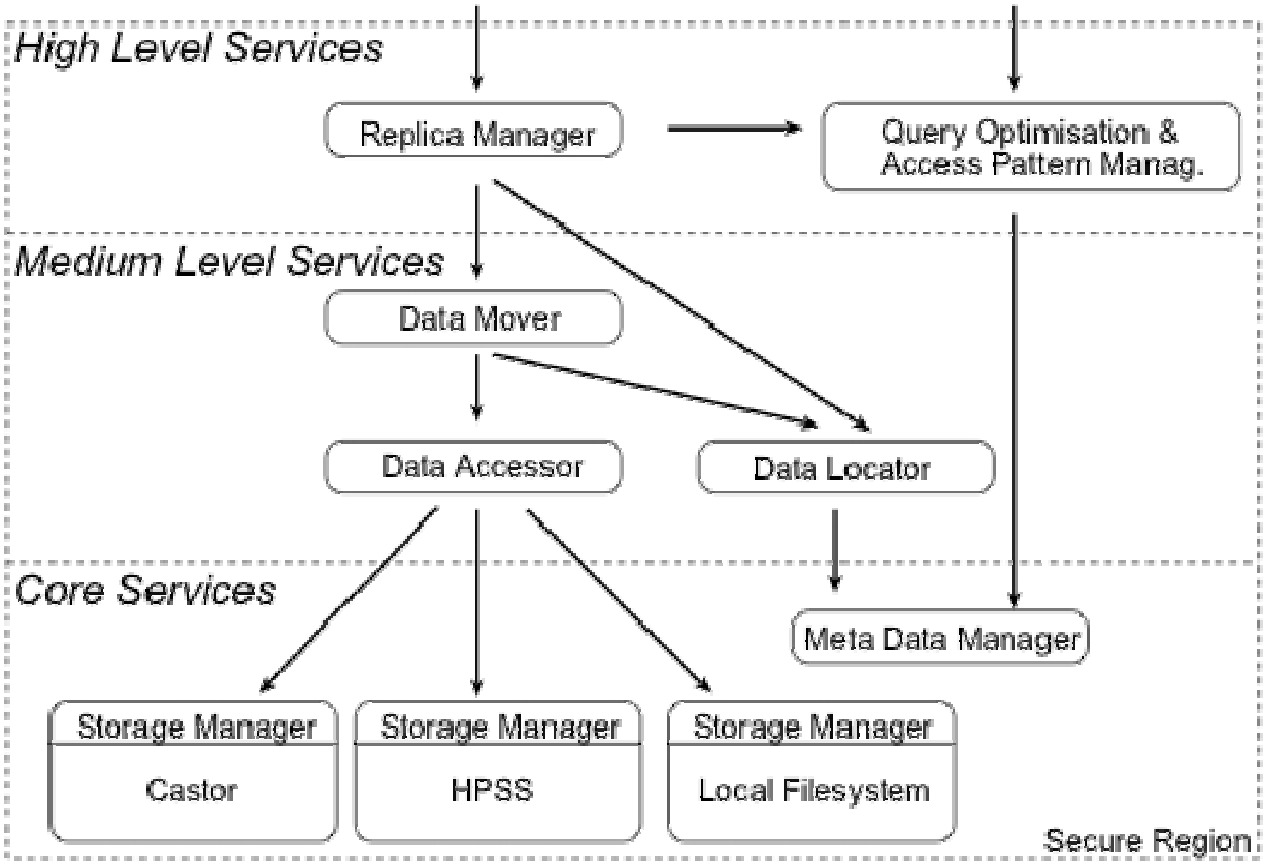
- improve efficiency of data analysis by integrating widely distributed processing power and data storage

Architecture



- easy to understand
- flexible: layered interfaces
- rapid prototyping: leverage previous work
- scalable
- respect distributed development: clearly defined and loosely coupled

Data Management Overview



Data Accessor

- must access a variety of storage systems
  - initial work focuses on HSM and file system
- converts Grid data access requests into something underlying storage will understand
  - also prepares underlying storage to deliver data
- hides complexities of data access from higher levels

Replication

- caching strategy, multiple identical files are stored in multiple locations
- provides faster access, better fault tolerance, better availability of data
- updates must be synchronized with all replicas
- replication problem involves:
  - how to physical transfer data, synchronization
  - deciding policies of when to trigger replica creation
- policies are not decided by a single entity
  - must provide services for task schedulers, Grid admin, local resource managers to replicate, maintain consistency, obtain information about replicas

Replication Manager

- users requests for data are routed through the Replication Manager
- “intelligent service”
  - analyzes access patterns, knows about distribution of files
- optimizes wide-area throughput via Grid cache
- Data Locator maps location independent name to location dependent name
- Data Accessor access files selected by Replication Manager

Meta Data

- catalogues of names and locations of files
- monitoring information
- grid configuration information
- policies enabling flexible and dynamic steering

- service is built on LDAP
  - fully distributed, hierarchical, versatile, uniform

Security

- site that owns data must ensure that sites hosting replicas provide same level of security
  - different sites = different security infrastructure
- synchronous update of replicas
  - more dangerous than on-demand or scheduled
  - better consistency and responsiveness
- consider security in replica selection
  - select from more “friendly” nodes
- differences between data and meta data

Provide flexibility for sites, not common policy.

Query Optimization

- goal is to select replica that will be cheapest to access

- considerations:
  - size of file
  - load on data server
  - method/protocols of access
  - bandwidth, distance, traffic
  - policies on remote access

MySRB & SRB

Distributed Data Collections

- single name space for data on multiple storage systems
  - support attributes associated with each registered data entity
  - handle multiple types of platforms
  - seamless access

Digital Libraries

- integrate remote archival storage systems, provide discovery and manipulation services
  - seamless authentication, single sign on
  - virtual organization structure
    - data organized into context-dependent structure
  - scale with increased dataset size

Persistent Archives

- support the migration of data collections onto new technologies, while preserving the ability to organize, discover, and access data
  - replication of data (little effort by users)
  - version control
  - access control at multiple levels, auditing

SRB

- client-server middleware
- provides means to organize data from multiple heterogeneous systems into one logical collection
- access data by attribute, not location
  - yields location transparency
- also supports:
  - replica storing, authentication, access control, auditing access, metadata

SRB

- federated server system
  - each SRB server manages a set of storage resources

- advantages:
  - location transparency
  - reliability and availability (replicas)
  - administrative reasons (different security protocols)
  - fault tolerance (automatic redirect to replicas)
  - integrated data access (can access backups, etc.,)
  - persistence (can easily move data to new resources)

MySRB

- web-based interface to SRB

- primary functionalities
  - collection and file management
  - metadata handling
  - access and display of files and metadata
    - browsing, search and query

MySRB: Data Movement

- ingest a file
  - user specifies a logical resource or a container
  - specifies any required and user-defined meta data

- register an object
  - no physical copy of file is in SRB
  - pointer to physical copy is stored
    - file in a file system
    - directory in a file system
    - SQL query
    - URL
    - method object or virtual data

MySRB: Data Movement

- replicate
  - any ingested or registered file
  - user specifies resource to hold replica
  - replica inherits all metadata
  - globally unique replica number returned

- register replicate / ingest replicate
  - register a new object as a semantically equal replica of an existing object

MySRB: Data Movement

- copy
  - creates copy of an object or registered object
  - copy is NOT replica of original
  - user-defined meta data is not copied
  - user specifies new resource, path name and collection for copy

- move
  - files and sub-collections may be moved
  - user-defined meta data does not change
  - ingested files may be physically moved

MySRB: Data Movement

- link
  - similar to soft linking in Unix
  - access control of original is used
  - original meta data can be viewed but not edited
  - chaining is not allowed (will point to original)

- delete
  - deletion of registered items will not physically delete
  - replicas are deleted one at a time, meta data is maintained until all removed
  - deleting a link = unlinking

- lock, pin, checkout
  - shared lock: user can edit, others can read
  - exclusive lock: only user can edit and read
  - pin: prevents deletion
  - checkout/checkin: rudimentary version control

MySRB: MetaData

- system-defined
  - created and maintained by SRB system
  - user can view and search on it
- user-defined
  - on entry, after entry with insert, copied from another object, extracted from object
- type-oriented
  - pre-defined sets of metadata associated with a type
- file-based
  - meta data is stored in a file, associated with object
- annotations
  - free form