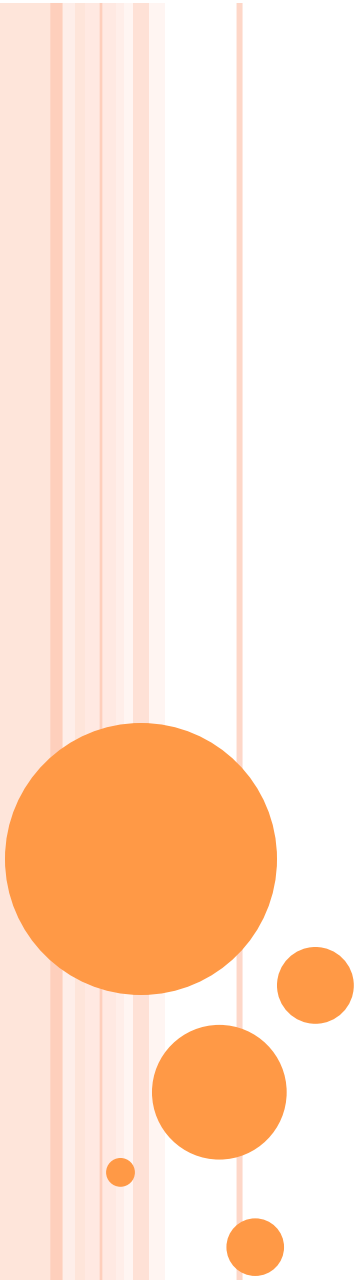


Grid Data Management

Data Management

- Distributed community of users need to access and analyze large amounts of data

Fusion community's International ITER project

- Requirement arises in both simulation and experimental science

Open Science Grid

DATA MANAGEMENT

- Huge raw volume of data
  - Measured in terabytes, petabytes, and further …
  - Data sets can be partitioned as small number of large files or large number of small files
  - Store it long term in appropriate places (e.g., tape silos)
  - Move input to where your job is running
  - Move output data from where your job ran to where you need it (eg. your workstation, long term storage)

Data Management on the Grid

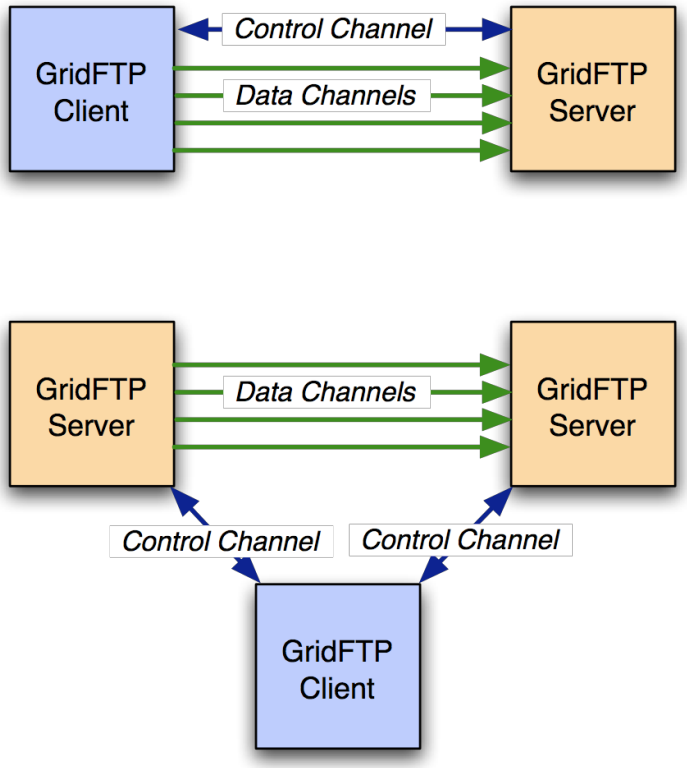
- How to move data/files to where I want?
  - GridFTP
- Data sets replicated for reliability and faster access
- Files have logical names
- Service that maps logical file names to physical locations
  - Replica Location Service (RLS)
  - Where are the files I want?

GRIDFTP

- High performance, secure, and reliable data transfer protocol based on the standard FTP
  - http://www.ogf.org/documents/GFD.20.pdf\
  - Multiple implementations exist, we'll focus on Globus GridFTP
- Globus GridFTP Features include
  - Strong authentication, encryption via Globus GSI
  - Multiple transport protocols - TCP, UDT
  - Parallel transport streams for faster transfer
  - Cluster-to-cluster or striped data movement
  - Multicasting and overlay routing
  - Support for reliable and restartable transfers

Basic Definitions

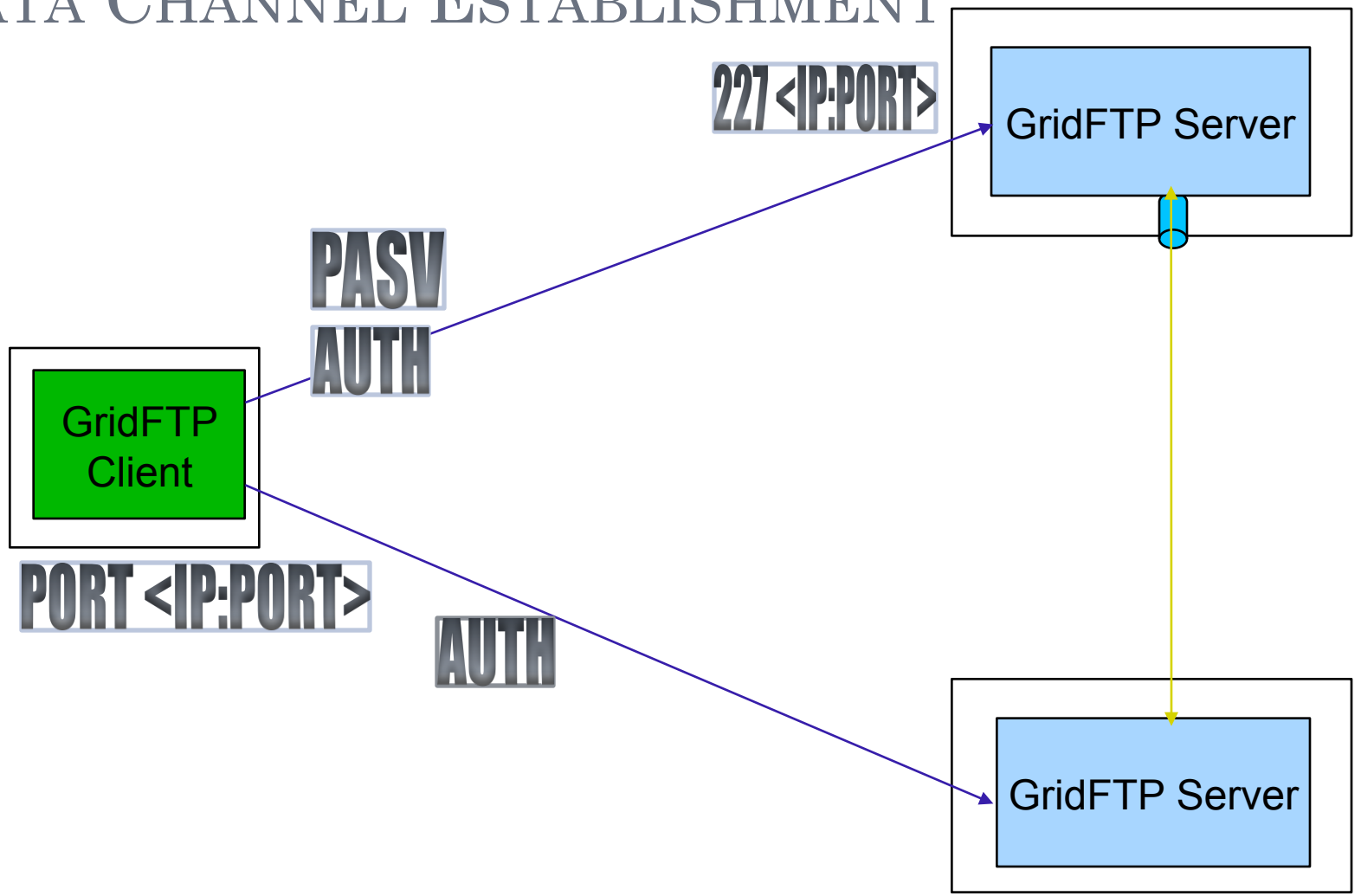
- Control Channel
  - TCP link over which commands and responses flow
  - Low bandwidth; encrypted and integrity protected by default
- Data Channel
  - Communication link(s) over which the actual data of interest flows
  - High Bandwidth; authenticated by default; encryption and integrity protection optional



Control Channel Establishment

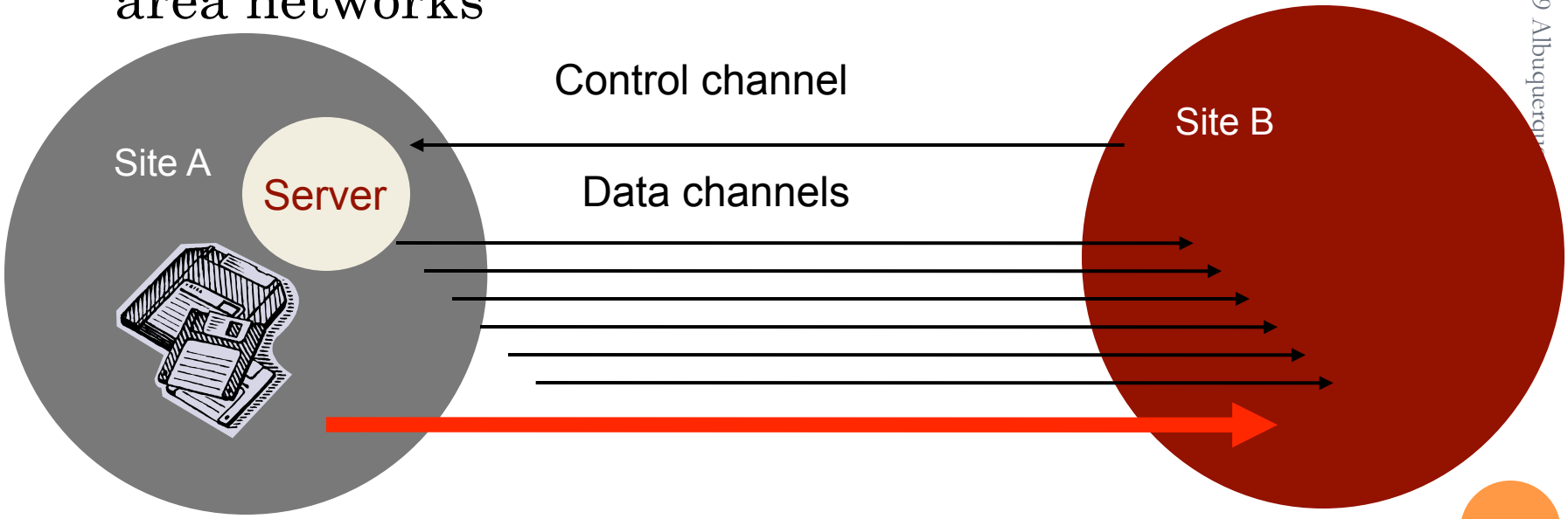
- Server listens on a well-known port (2811)
- Client form a TCP Connection to server
- Authentication
  - Anonymous
  - Clear text USER <username>/PASS <pw>
  - Base 64 encoded GSI handshake

Data Channel Establishment

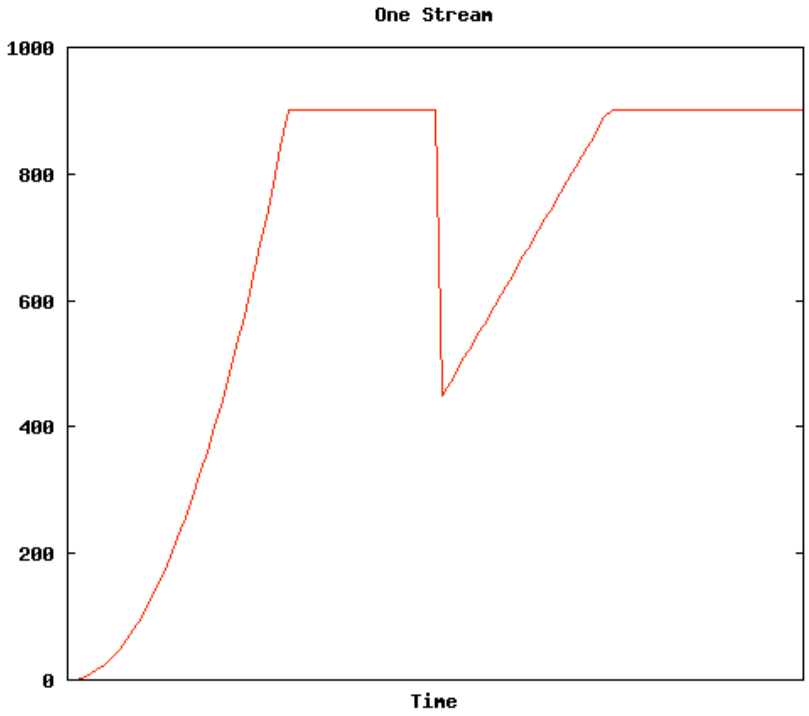


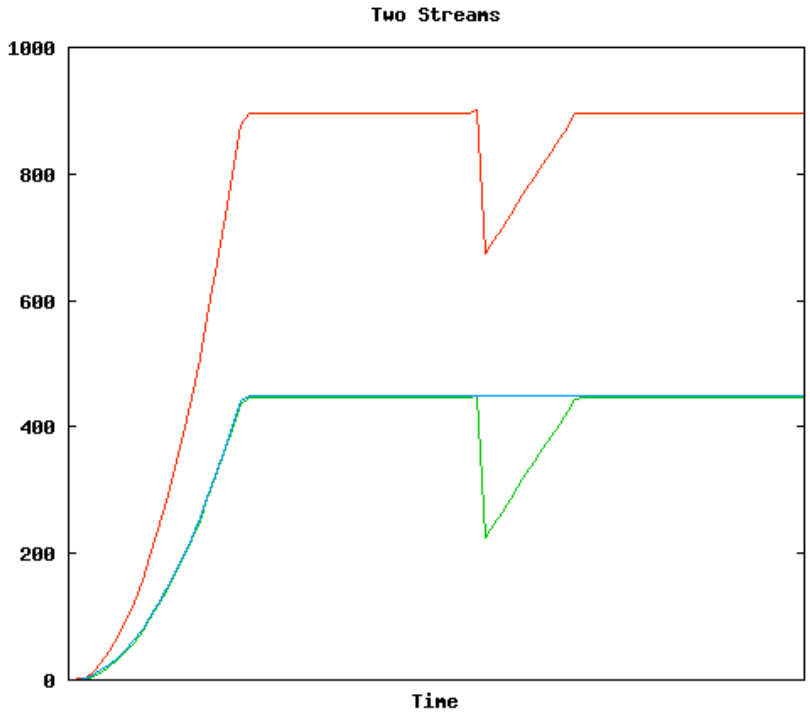
GOING FAST – PARALLEL STREAMS

- Use several data channels
- TCP - default transport protocol used by GridFTP
- TCP has limitations on high bandwidth wide area networks

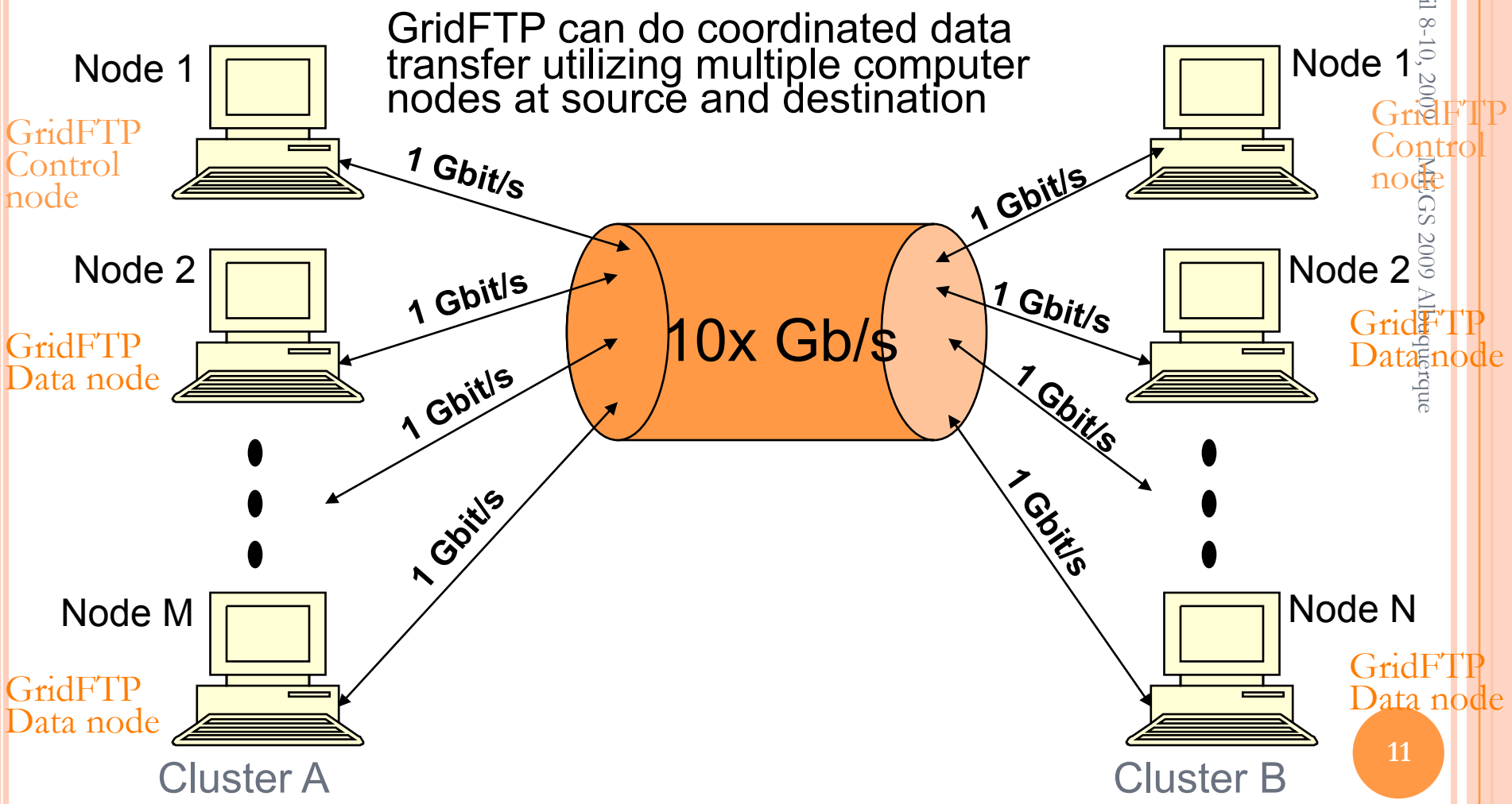


Parallel Streams





CLUSTER-TO-CLUSTER DATA TRANSFER



Open Science Grid

GRIDFTP USAGE

- **globus-url-copy - commonly used GridFTP client**
  - **Usage: globus-url-copy [options] srcurl dsturl**
- Conventions on URL formats:
  - `file:///home/YOURLOGIN/dataex/largefile`
    - a file called **largefile** on the local file system, in directory **/home/YOURLOGIN/dataex/**
  - `gsiftp://osg-edu.cs.wisc.edu/scratch/YOURLOGIN/`
    - a directory accessible via gsiftp on the host called `osg-edu.cs.wisc.edu` in directory **/scratch/YOURLOGIN**.

GRIDFTP TRANSFERS USING GLOBUS-URL-COPY

- **globus-url-copy**

  file:///home/YOURLOGIN/dataex/myfile

  gsiftp://osg-edu.cs.wisc.edu/nfs/osgedu/YOURLOGIN/ex1

- **globus-url-copy**

gsiftp://osg-edu.cs.wisc.edu/nfs/osgedu/YOURLOGIN/ex2

gsiftp://tp-osg.ci.uchicago.edu/YOURLOGIN/ex3

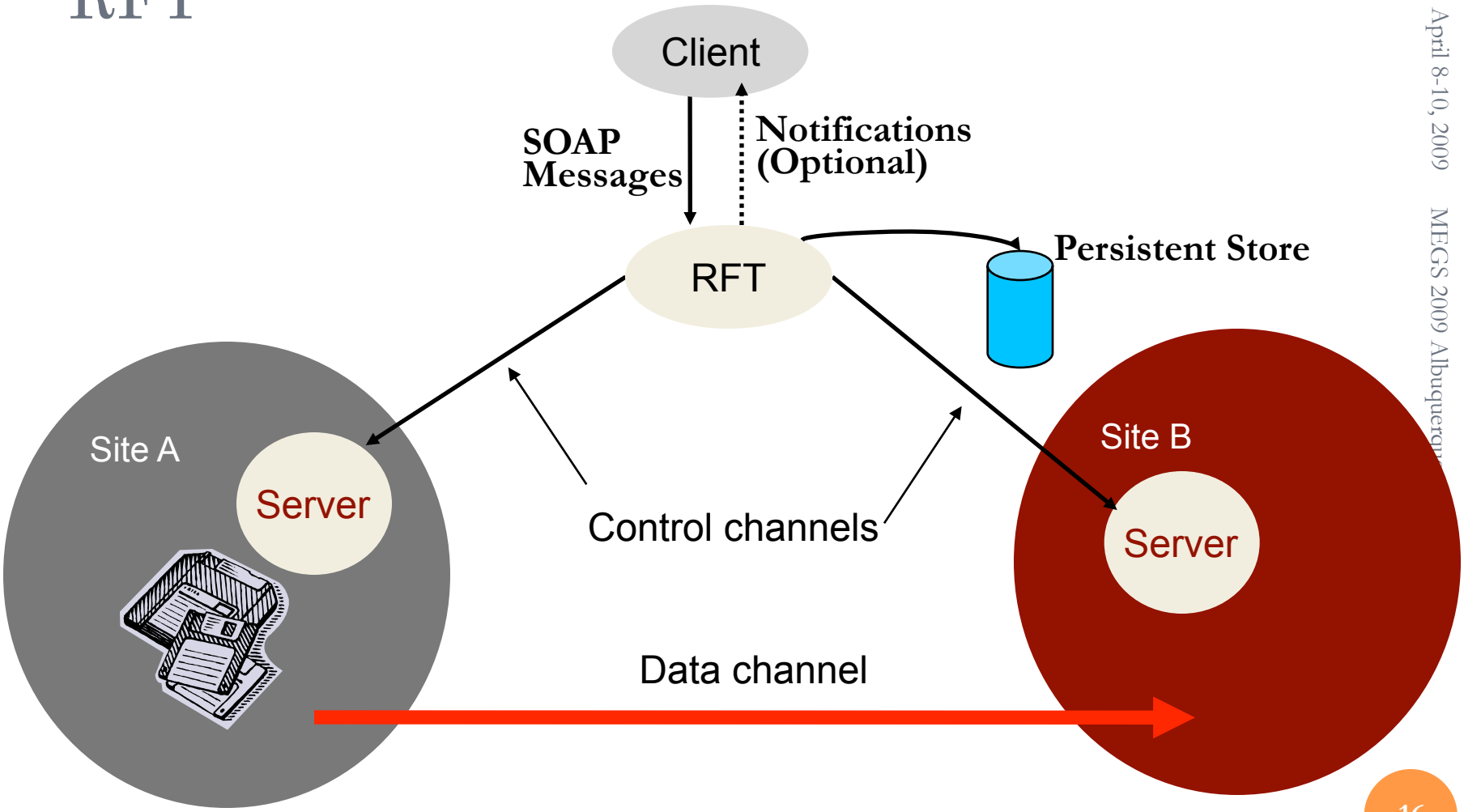
HANDLING FAILURES

- GridFTP server sends restart and performance markers periodically
  - Restart markers are helpful if there is any failure
  - No need to transfer the entire file again
  - Use restart markers and transfer only the missing pieces
- GridFTP supports partial file transfers
  - Globus-url-copy has a retry option
  - Recover from transient server and network failures
  - What if the client (globus-url-copy) fails in the middle of a transfer?

RFT = Reliable file transfer

- GridFTP client that provides more reliability and fault tolerance for file transfers
  - Part of the Globus Toolkit
- RFT acts as a client to GridFTP, providing management of a large number of transfer jobs (same as Condor to GRAM)
- RFT can
  - keep track of the state of each job
  - run several transfers at once
  - deal with connection failure, network failure, failure of any of the servers involved.

RFT


Client
SOAP Messages
Notifications (Optional)
RFT
Persistent Store
Site A
Server
Control channels
Site B
Server
Data channel

Storage and Grid

- Unlike compute/network resources, storage resources are not available when jobs are done
- Release resource usage when done, unreleased resource need to be garbage collected
- Need to enforce quotas
- Need to ensure fairness of space allocation and scheduling

What is SRM?

- Storage Resource Managers (SRMs) are middleware components
  - whose function is to provide
    - dynamic space allocation
    - file management on shared storage resources on the Grid
  - Different **implementations** for underlying storage systems are based on the same SRM **specification**

MANAGING SPACES

- Negotiation
  - Client asks for space: Guaranteed_C, MaxDesired
  - SRM return: Guaranteed_S <= Guaranteed_C, best effort <= MaxDesired
- Types of spaces
  - Access Latency (Online, Nearline)
  - Retention Policy (Replica, Output, Custodial)
  - Subject to limits per client (SRM or VO policies)
- Lifetime
  - Negotiated: Lifetime_C requested
  - SRM return: Lifetime_S <= Lifetime_C
- Reference handle
  - SRM returns space reference handle (space token)
- Updating space
  - Resize for more space or release unused space
  - Extend or shorten the lifetime of a space

File Management

- Assignment of files to spaces
  - Files can be assigned to any space, provided that their lifetime is shorter than the remaining lifetime of the space
  - Files can be put into an SRM without explicit reservation
  - Default spaces are not visible to client
- Files already in the SRM can be moved to other spaces
  - By srmChangeSpaceForFiles
- Files already in the SRM can be pinned in spaces
  - By requesting specific files (srmPrepareToGet)
  - By pre-loading them into online space (srmBringOnline)
- Releasing files from space by a user
  - Release all files that user brought into the space whose lifetime has not expired

TRANSFER PROTOCOL NEGOTIATION

- Negotiation
  - Client provides an ordered list of preferred transfer protocols
  - SRM returns first protocol from the list it supports
  - Example
    - Client provided protocols list: bbftp, gridftp, ftp
    - SRM returns: gridftp
- Advantages
  - Easy to introduce new protocols
  - User controls which transfer protocol to use
- How it is returned?
  - Transfer URL (TURL)
  - Example: bbftp://dm.slac.edu//temp/run11/File678.txt

Site URL and Transfer URL

- Provide: Site URL (SURL)
  - URL known externally – e.g. in Replica Catalogs
  - e.g. srm://ibm.cnaf.infn.it:8444/dteam/test.10193
- Get back: Transfer URL (TURL)
  - Path can be different from SURL – SRM internal mapping
  - Protocol chosen by SRM based on request protocol preference
  - e.g. gsiftp://ibm139.cnaf.infn.it:2811//gpfs/sto1/dteam/test.10193
- One SURL can have many TURLs
  - Files can be replicated in multiple storage components
  - Files may be in near-line and/or on-line storage
  - In a light-weight SRM (a single file system on disk)
    - SURL may be the same as TURL except protocol

DIRECTORY MANAGEMENT

- Usual unix semantics
  - srmLs, srmMkdir, srmMv, srmRm, srmRmdir
- A single directory for all spaces
  - No directories for each file type
  - File assignment to spaces is virtual

OSG & DATA MANAGEMENT

- OSG relies on GridFTP protocol for the raw transport of the data using Globus GridFTP in all cases except where interfaces to storage management systems (rather than file systems) dictate individual implementations.
- OSG supports the SRM interface to storage resources to enable management of space and data transfers to prevent unexpected errors due to running out of space, to prevent overload of the GridFTP services, and to provide capabilities for pre-staging, pinning and retention of the data files. OSG currently provides reference implementations of two storage systems the (BeStMan) and dCache

STORAGE SOFTWARE USED ON OSG

- dCache
- Bestman with various backends (Lustre, Xrootd, hdfs, gpfs, cluster file system)

dCache

- Allows space on nodes to be aggregated into a single namespace
- Does not present a posix compliant file system yet
- Supports tape backup systems for migrating infrequently used data to tape for nearline storage

Birds Home

dCache

Disk Space Usage

CellName	DomainName	Total Space/MB	Free Space/MB	Precious Space/MB	Layout (precious/used/free)
uct3-c001_1	uct3-c001Domain	1906688	349027	1557660	
uct3-c002_1	uct3-c002Domain	1906688	429878	1469270	
uct3-c003_1	uct3-c003Domain	1906688	429750	1469486	
uct3-c004_1	uct3-c004Domain	1906688	429430	1469872	
uct3-c005_1	uct3-c005Domain	1906688	429781	1471571	
uct3-c006_1	uct3-c006Domain	1906688	438487	1463894	
uct3-c007_1	uct3-c007Domain	1906688	586994	1312239	
uct3-c008_1	uct3-c008Domain	1906688	419113	1480136	
uct3-c009_1	uct3-c009Domain	1906688	429732	1470595	
uct3-c010_1	uct3-c010Domain	1906688	429773	1469146	
uct3-c011_1	uct3-c011Domain	1906688	617364	1283145	
uct3-c012_1	uct3-c012Domain	1906688	438581	1462039	
uct3-c013_1	uct3-c013Domain	1906688	429768	1470707	
uct3-c014_1	uct3-c014Domain	1906688	429650	1471442	
uct3-c015_1	uct3-c015Domain	1906688	616915	1282996	
uct3-c016_1	uct3-c016Domain	1906688	226563	1680043	
uct3-c017_1	uct3-c017Domain	1906688	429690	1470719	
uct3-c018_1	uct3-c018Domain	1906688	429474	1477213	
uct3-c019_1	uct3-c019Domain	1906688	516282	1382958	
uct3-c020_1	uct3-c020Domain	1906688	429480	1471125	
uct3-c021_1	uct3-c021Domain	1906688	436751	1462366	
uct3-edge1_1	uct3-edge1Domain	1429504	429276	992844	
uct3-edge2_1	uct3-edge2Domain	1906688	398997	1500959	

DCACHE POOL USAGE

Space usage for the dCache pool on the tier 3 cluster at the University of Chicago. Each node contributes about 2TB of space

uct3-edge3.uchicago.edu

Help Birds Home

dCache Statistics of dCache Domain : uct3-edge3.uchicago.edu

Year	Absolute Values	Data / MBytes	Relative Values
2009		34.631.019 + 0 + 194.428	
		34.646.768	
		142.746.204 + 0	
2008		38.800.700 + 0 + 22.176.128	
		34.631.019	
		550.954.555 + 0	
2007		25.511.771 + 0 + 7.467.684	
		38.800.700	
		32.637.825 + 0	

DCACHE STATISTICS

Annual statistics from the dCache pool at the tier 3 cluster at the University of Chicago.

BESTMAN

- Bestman provides a srm gateway on top of a posix file system
- Commonly used backends include (xrootd, hdfs, lustre, gpfs)
- Simpler to configure and maintain then dCache
- Doesn't have all of the flexibility that dCache has

Comparison between Bestman and dCache

- dCache has better support for some srm features (dynamic space reservation, replica management) and supports using a tape library as a backing store
- dCache currently doesn't have have a filesystem with posix semantics (need to use srm commands or dccp)
- Bestman is easier to setup and maintain
- Because Bestman is a shim layer on a filesystem, you can use your own solution as the backend filesystem (xrootd, hdfs, gpfs, lustre, etc.)
- Both systems require several servers in order to have a production resource

dCache and Bestman usage on OSG

- Primarily used by larger VOs
- Heavy usage by CMS, ATLAS, CDF/Dzero
- CMS is using dCache primarily but has been moving to bestman/xrootd or bestman/hdfs implementations
- Tier 1 facilities (Fermilab and Brookhaven) are using dCache with tape libraries
- Other resources use a mix of dCache and Bestman

CREDITS

Bill Allcock allcock@mcs.anl.gov
Ben Clifford benc@ci.uchicago.edu
Scott Koranda skoranda@uwm.edu
Alex Sim asim@lbl.gov

Directory Discovery

- Notice anything odd when creating the directory on the clemson resource (osgce)?
  - `condor_exec.exe: cannot create directory `/nfs/osgedu/train20': No such file or directory`
- The resource changed the location of the directory!
- Problem with using fixed locations

A Solution

- Luckily, the location of the data directory is available in an environment variable:

```
train20@vm-125-58:~/$ globus-job-run
  osgce.cs.clemson.edu/jobmanager-fork /bin/env | grep
  OSG_DATA
OSG_DATA=/export/osg/data
```

- Let's run the command again
  - `globus-job-run osgce.cs.clemson.edu/jobmanager-fork /bin/mkdir /export/osg/osgedu/YOURNAME`