Grid Data Management

Open Science Grid

Data Management

- Want to move data around:
  - 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)
- Exercises will introduce Globus Toolkit's component called GridFTP

High-performance tools needed to solve several data problems.

- The huge raw volume of data:
  - Storing it
  - Moving it
  - Measured in terabytes, petabytes, and further …
- The huge number of filenames:
  - 10^{12} filenames is expected soon
  - Collection of 10^{12} of anything is a lot to handle efficiently
- How to find the data

Data Questions on the Grid

Questions for which you want Grid tools to address

- Where are the files I want?
- How to move data/files to where I want?

GridFTP

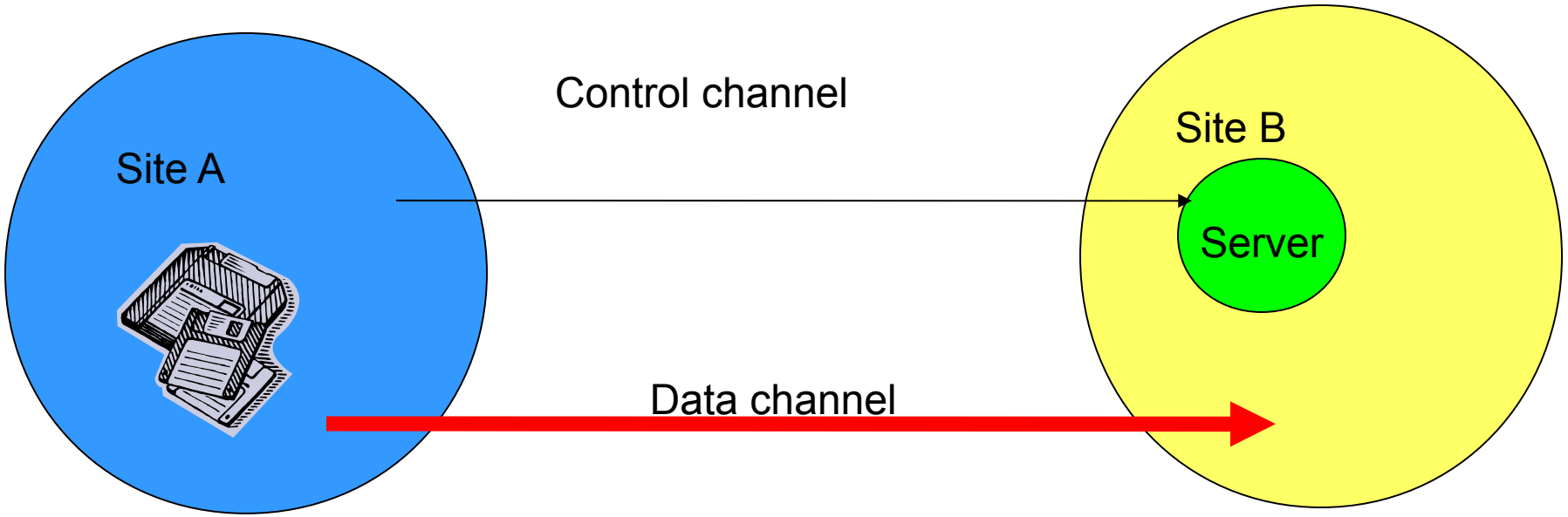
- high performance, secure, and reliable data transfer protocol based on the standard FTP
  - http://www.ogf.org/documents/GFD.20.pdf
- Extensions include
  - Strong authentication, encryption via Globus GSI
  - Multiple data channels for parallel transfers
  - Third-party transfers
  - Tunable network & I/O parameters
  - Authenticated reusable channels
  - Server side processing, command pipelining

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

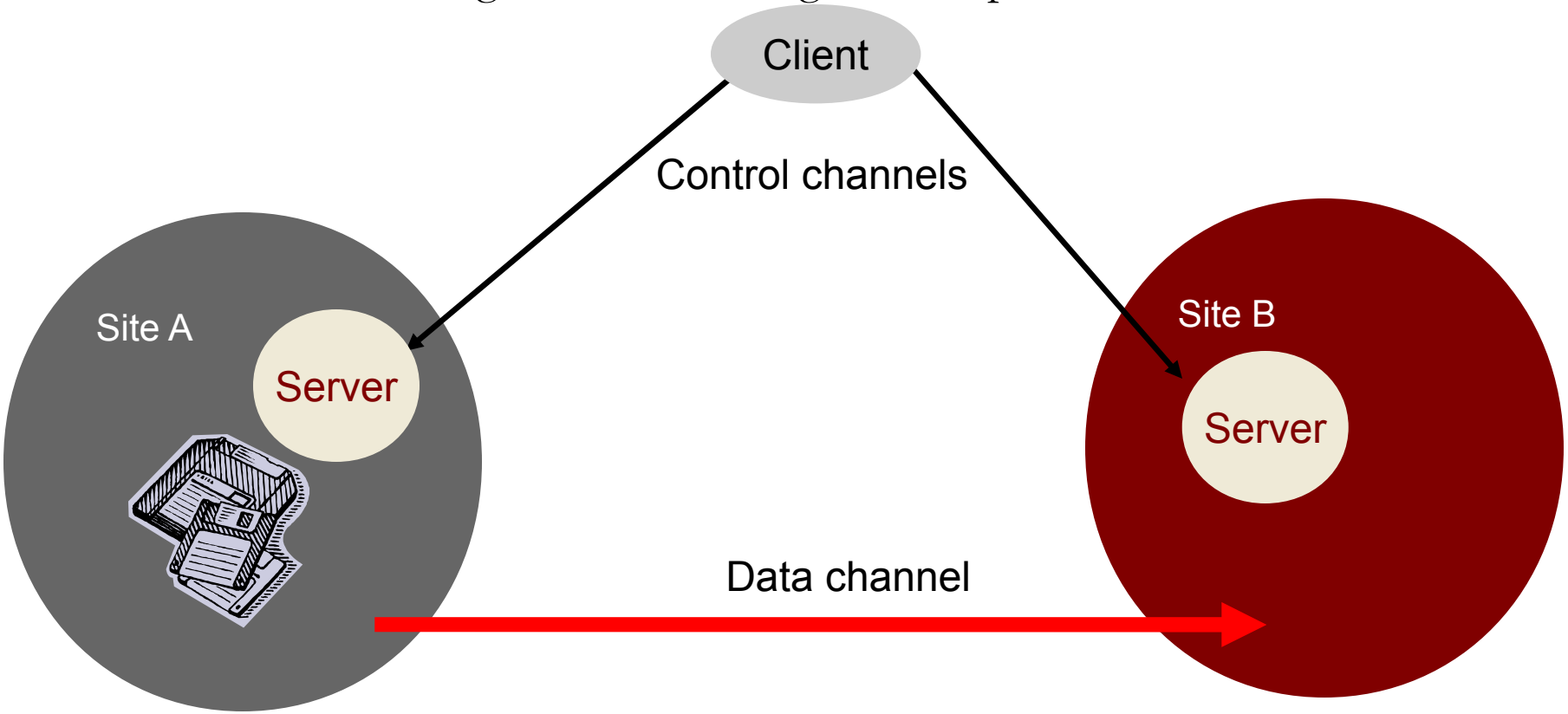
A file transfer with GridFTP

- Control channel can go either way
  - Depends on which end is client, which end is server
- Data channel is still in same direction



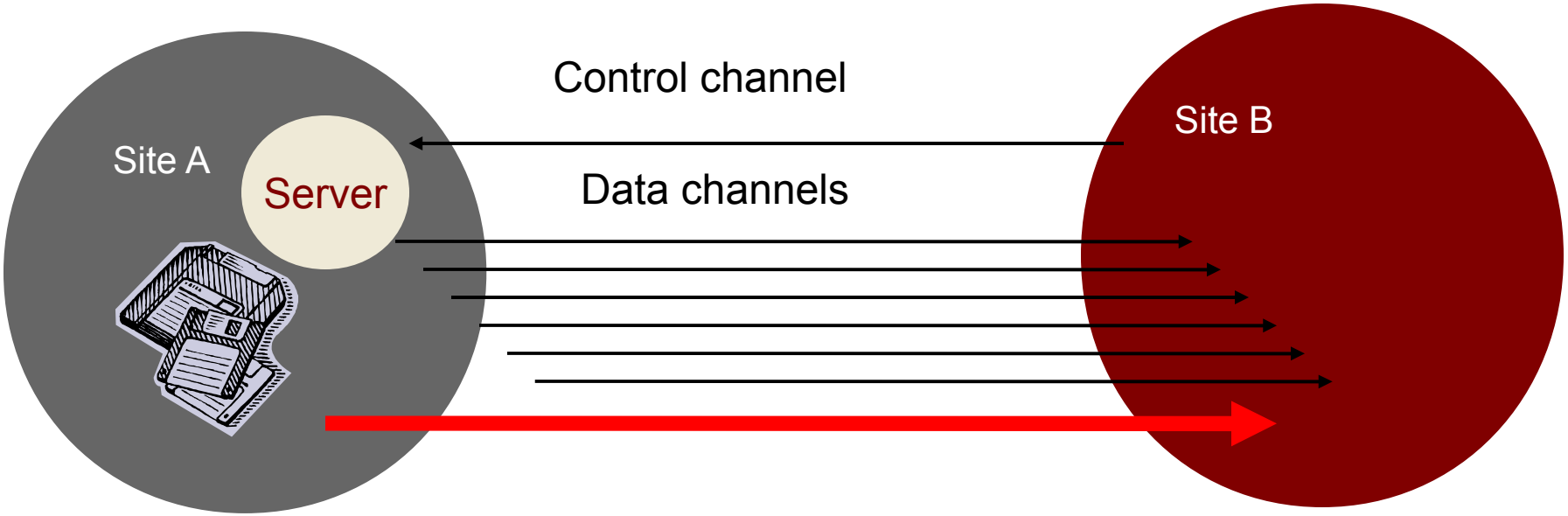
Third party transfer

- File transfer without data flowing through the client.
- Controller can be separate from src/dest
- Useful for moving data from storage to compute



Going fast – parallel streams

- Use several data channels



To make GridFTP go really fast:

- Use fast disks/filesystems
  - Filesystem should read/write > 30 MB/second
- Configure TCP for performance
  - See the TCP Tuning Guide at http://www-didc.lbl.gov/TCP-tuning/
- Patch your Linux kernel with web100 patch
  - Important work-around for Linux TCP "feature"
  - See http://www.web100.org
- Understand your network path

GridFTP usage

- **globus-url-copy**
- Convensions 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 examples

- **globus-url-copy**
  file:///home/YOURLOGIN/dataex/myfile
  gsiftp://osg-edu.cs.wisc.edu/nfs/osgedu/YOURLOGIN/ex1

- **globus-url-copy**

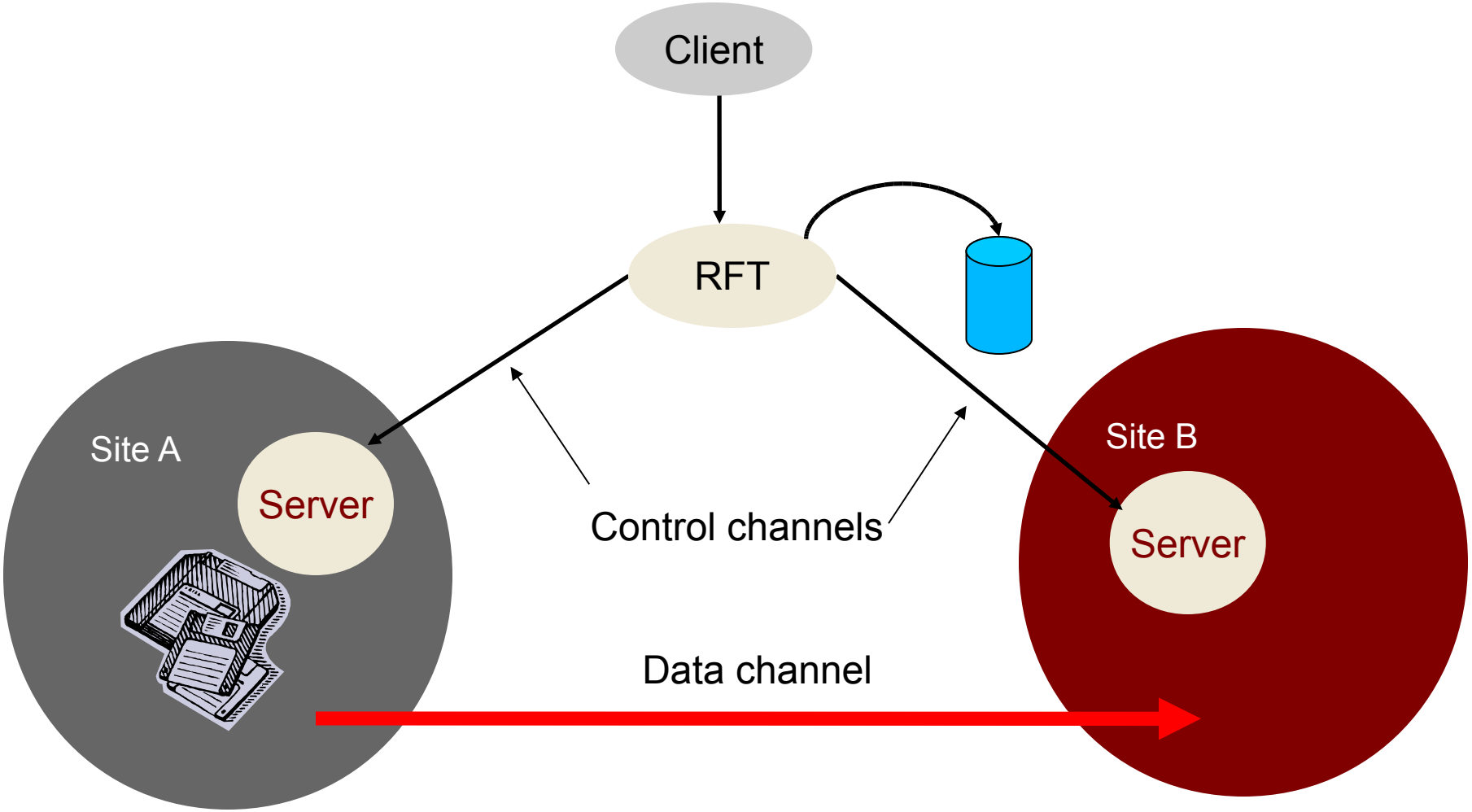
gsiftp://osg-edu.cs.wisc.edu/nfs/osgedu/YOURLOGIN/ex2

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

RFT = Reliable file transfer

- protocol that provides 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
RFT
Site A
Server
Control channels
Site B
Server
Data channel

RFT

- WS-RF compliant High Performance data transfer service
  - Soft state
  - Notifications/Query
- Reliability on top of high performance provided by GridFTP
  - Fire and Forget
  - Integrated Automatic Failure Recovery
    - Network level failures
    - System level failures, etc.

RFT example

- Use the rft command with a .xfr file
- cp /soft/globus-4.0.3-r1/share/globus_wsrf_rft_client/transfer.xfr rft.xfr
- Edit rft.xfr to match your needs
- rft -h terminable.ci.uchicago.edu -f ./rft.xfr

RLS -Replica Location Service

- RLS
  - component of the data grid architecture (Globus component)
  - It provides access to mapping information from logical names to physical names of items
  - Its goal is to

    reduce access latency, improve data locality, improve robustness, scalability and performance for distributed applications

- RLS produces replica catalogs (LRCs), which represent mappings between logical and physical files scattered across the storage system.
  - For better performance, the LRC can be indexed.

RLS -Replica Location Service

- RLS maps logical filenames to physical filenames.
- Logical Filenames (LFN)
  - Names a file with interesting data in it
  - Doesn't refer to location (which host, or where in a host)
- Physical Filenames (PFN)
  - Refers to a file on some filesystem somewhere
  - Often use `gsiftp://` URLs to specify
- Two RLS catalogs:
  - Local Replica Catalog (LRC) and
  - Replica Location Index (RLI)

Local Replica Catalog (LRC)

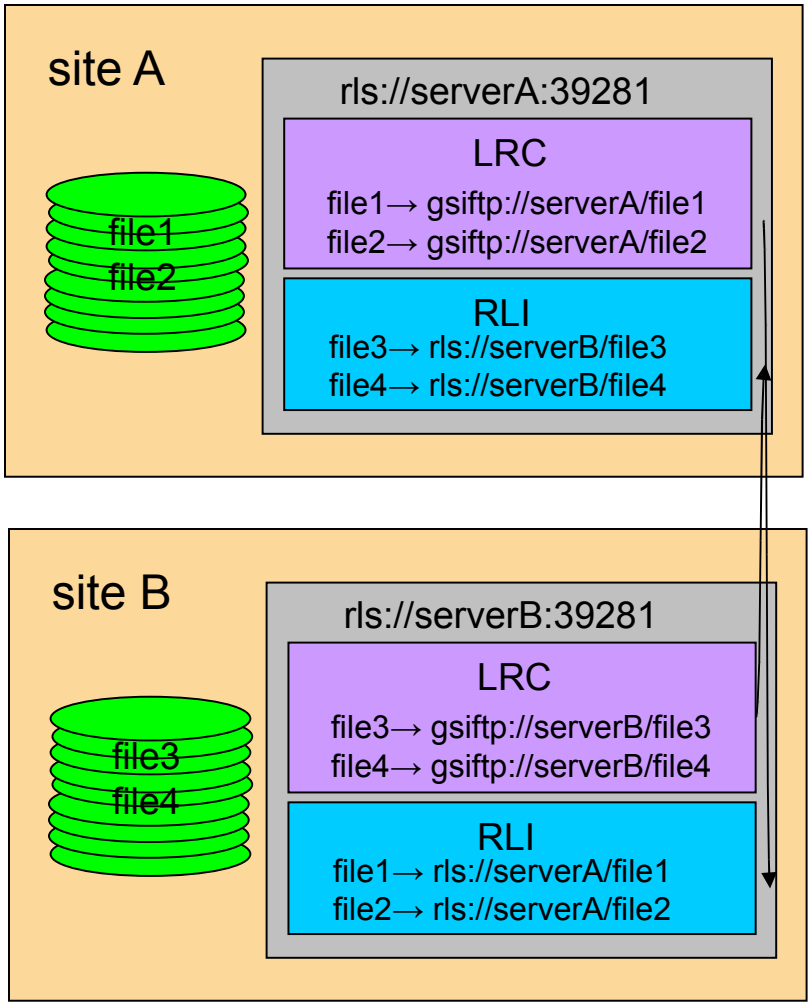
- stores mappings from LFNs to PFNs.
- Interaction:
  - *Q:* Where can I get filename 'experiment_result_1'?
  - *A:* You can get it from

    gsiftp://gridlab1.ci.uchicago.edu/home/benc/r.txt
- Undesirable to have one of these for whole grid
  - Lots of data
  - Single point of failure

Replica Location Index (RLI)

- stores mappings from LFNs to LRCs.
- Interaction:
  - *Q:* Who can tell me about filename 'experiment_result_1'.
  - *A:* You can get more info from the LRC at gridlab1
  - (Then go to ask that LRC for more info)
- Failure of one RLI or LRC doesn't break everything
- RLI stores reduced set of information, so can cope with many more mappings

Globus RLS


site A
rls://serverA:39281
LRC
file1→ gsiftp://serverA/file1
file2→ gsiftp://serverA/file2
RLI
file3→ rls://serverB/file3
file4→ rls://serverB/file4
file1
file2
site B
rls://serverB:39281
LRC
file3→ gsiftp://serverB/file3
file4→ gsiftp://serverB/file4
RLI
file1→ rls://serverA/file1
file2→ rls://serverA/file2
file3
file4

Globus RLS

- Quick Review
  - LFN → logical filename (think of as simple filename)
  - PFN → physical filename (think of as a URL)
  - LRC → your local catalog of maps from LFNs to PFNs
    - H-R-792845521-16.gwf → gsiftp://dataserver.phys.uwm.edu/LIGO/H-R-792845521-16.gwf
  - RLI → your local catalog of maps from LFNs to LRCs
    - H-R-792845521-16.gwf → LRCs at MIT, PSU, Caltech, and UW-M
  - LRCs inform RLIs about mappings known
- Can query for files is a 2-step process: find files on your Grid by
  - querying RLI(s) to get LRC(s)
  - then query LRC(s) to get URL(s)

Globus RLS: Server Perspective

- Mappings LFNs → PFNs kept in database
  - Uses generic ODBC interface to talk to any (good) RDBM
  - MySQL, PostgreSQL, Oracle, DB2,...
  - All RDBM details hidden from administrator and user
    - well, not quite
    - RDBM may need to be "tuned" for performance
    - but one can start off knowing very little about RDBMs

Globus RLS: Server Perspective

Mappings LFNs → LRCs stored in 1 of 2 ways

- **table in database**
  - full, complete listing from LRCs that update your RLI
  - requires each LRC to send your RLI full, complete list
    - as number of LFNs in catalog grows, this becomes substantial
    - 10^8 filenames at 64 bytes per filename ~ 6 GB
- **in memory in a special hash called Bloom filter**
  - 10^8 filenames stored in as little as 256 MB
    - easy for LRC to create Bloom filter and send over network to RLIs
  - can cause RLI to lie when asked if knows about a LFN
    - only false-positives
    - tunable error rate
    - acceptable in many contexts
  - Wild carding not possible with Bloom Filters

RLS command line tools

- **globus-rls-admin**
  - administrative tasks
    - ping server
    - connect RLIs and LRCs together
- **globus-rls-cli**
  - end user tasks
    - query LRC and RLI
    - add mappings to LRC

Globus RLS: Client Perspective

Two ways for clients to interact with RLS Server

- **globus-rls-cli** simple command-line tool
  - query
  - create new mappings
- "roll your own" client by coding against API
  - Java
  - C
  - Python

Globus-rls-cli

Simple query to LRC to find a PFN for LFN

- Note more then one PFN may be returned

```
$ globus-rls-cli query lrc lfn some-file.jpg rls://dataserver:39281

some-file.jpg : file://localhost/netdata/s001/S1/R/H/714023808-
  714029599/some-file.jpg
some-file.jpg : file://medusa-
  slave001.medusa.phys.uwm.edu/data/S1/R/H/714023808-714029599/some-
  file.jpg
some-file.jpg :
  gsiftp://dataserver.phys.uwm.edu:15000/data/gsiftp_root/cluster_storage/
  data/s001/S1/R/H/714023808-714029599/some-file.jpg
```

- Server and client sane if LFN not found

```
$ globus-rls-cli query lrc lfn foo rls://dataserver
LFN doesn't exist: foo
$ echo $?
1
```

Globus-rls-cli

Wildcard searches of LRC supported

- probably a good idea to quote LFN wildcard expression

```
$ globus-rls-cli query wildcard lrc lfn H-R-7140242*-16.gwf
   rls://dataserver:39281
  H-R-714024208-16.gwf:
   gsiftp://dataserver.phys.uwm.edu:15000/data/gsiftp_root/cluster_stor
   age/data/s001/S1/R/H/714023808-714029599/H-R-714024208-16.gwf
  H-R-714024224-16.gwf:
   gsiftp://dataserver.phys.uwm.edu:15000/data/gsiftp_root/cluster_stor
   age/data/s001/S1/R/H/714023808-714029599/H-R-714024224-16.gwf
```

Globus-rls-cli

Bulk queries also supported

- obtain PFNs for more then one LFN at a time

```
$ globus-rls-cli bulk query lrc lfn H-R-714024224-16.gwf
   H-R-714024320-16.gwf rls://dataserver
   H-R-714024320-16.gwf:
    gsiftp://dataserver.phys.uwm.edu:15000/data/gsiftp_root/
    cluster_storage/data/s001/S1/R/H/714023808-714029599/H-
    R-714024320-16.gwf
   H-R-714024224-16.gwf:
    gsiftp://dataserver.phys.uwm.edu:15000/data/gsiftp_root/
    cluster_storage/data/s001/S1/R/H/714023808-714029599/H-
    R-714024224-16.gwf
```

Globus-rls-cli

Simple query to RLI to locate a LFN -> LRC mapping

- then query that LRC for the PFN

```
$ globus-rls-cli query rli lfn example-file.gwf
  rls://dataserver

  example-file.gwf: rls://ldas-cit.ligo.caltech.edu:39281

$ globus-rls-cli query lrc lfn example-file.gwf rls://ldas-
  cit.ligo.caltech.edu:39281

  example-file: gsiftp://ldas-
  cit.ligo.caltech.edu:15000/archive/S1/L0/LHO/H-R-7140/H-R-
  714024224-16.gwf
```

Globus-rls-cli

- Bulk queries to RLI also supported

```
$ globus-rls-cli bulk query rli lfn H-R-714024224-16.gwf H-R-714024320-
   16.gwf rls://dataserver
  H-R-714024320-16.gwf: rls://ldas-cit.ligo.caltech.edu:39281
  H-R-714024224-16.gwf: rls://ldas-cit.ligo.caltech.edu:39281
```

- Wildcard queries to RLI may not be supported!
  - no wildcards when using Bloom filter updates

```
$ globus-rls-cli query wildcard rli lfn "H-R-7140242*-16.gwf"
   rls://dataserver
Operation is unsupported: Wildcard searches with Bloom filters
```

Globus-rls-cli

Create new LFN → PFN mappings

- use **`create`** to create 1st mapping for a LFN

```
$ globus-rls-cli create file1 gsiftp://dataserver/file1
   rls://dataserver
```

- use **`add`** to add more mappings for a LFN

```
$ globus-rls-cli add file1 file://dataserver/file1
   rls://dataserver
```

- use **`delete`** to remove a mapping for a LFN
  - when last mapping is deleted for a LFN the LFN is also deleted
  - cannot have LFN in LRC without a mapping

```
$ globus-rls-cli delete file1 file://file1 rls://dataserver
```

Globus-rls-cli

LRC can also store attributes about LFN and PFNs

- size of LFN in bytes?
- md5 checksum for a LFN?
- ranking for a PFN or URL?
- extensible...you choose attributes to create and add
- can search catalog on the attributes
- attributes limited to
  - strings
  - integers
  - floating point (double)
  - date/time

Globus-rls-cli

- Create attribute first then add values for LFNs

```
$ globus-rls-cli attribute define md5checksum lfn string
   rls://dataserver
$ globus-rls-cli attribute add file1 md5checksum lfn
   string 42947c86b8a08f067b178d56a77b2650 rls://dataserver
```

- Then query on the attribute

```
$ globus-rls-cli attribute query file1 md5checksum lfn
   rls://dataserver
  md5checksum: string: 42947c86b8a08f067b178d56a77b2650
```

Bloom filters

- LRC-to-RLI flow can happen in two ways:
  - LRC sends list of all its LFNs (but not PFNs) to the RLI. RLI stores whole list.
    - Answer accurately: "Yes I know" / "No I don't know"
    - Expensive to move and store large list
  - Bloom filters
    - LRC generates a Bloom filter of all of its LFNs
    - Bloom filter is a bitmap that is much smaller than whole list of LFNs
    - Answers less accurately: "Maybe I know" / "No I don't know". Might end up querying LRCs unnecessarily (but we won't ever get wrong answers)
    - can't do a wildcard search

Related Work

- Storage Resource Manager (SRM)
  - Equivalent of a job scheduler for storage; allocates space, makes sure it doesn't get swapped out before you are done (pinning); handles staging to and from tape
  - http://sdm.lbl.gov/indexproj.php?ProjectID=SRM
- dCache
  - provide a system for storing and retrieving huge amounts of data, distributed among a large number of heterogenous server nodes, under a single virtual filesystem tree with a variety of standard access methods.
  - http://www.dcache.org/
- BeSTMan

Related Work

- Globus Metadata Catalog
  - a stand-alone metadata catalog service with an OGSA service interface. The metadata catalog associates application-specific descriptions with data files, tables, or objects. These descriptions, which are encoded in structured ways defined by "schema" or community standards, make it easier for users and applications to locate data relevant to specific problems.
    - "I want the temperature, barometric pressure, and CO2 concentrations for this geographic area"
  - http://www.globus.org/grid_software/data/mcs.php

Related Work

- Stork
  - Cross between RFT and Condor DAGMAN
  - make data placement activities "first class citizens" in the Grid just like the computational jobs. They will be queued, scheduled, monitored, managed, and even check-pointed. More importantly, it will be made sure that they complete successfully and without any human interaction.
  - http://www.cs.wisc.edu/condor/stork/
- Storage Resource Broker
  - supports shared collections that can be distributed across multiple organizations and heterogeneous storage systems. The SRB can be used as a Data Grid Management System (DGMS) that provides a hierarchical logical namespace to manage the organization of data (usually files).
  - http://www.sdsc.edu/srb/index.php/Main_Page

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

Credits

Bill Allcock allcock@mcs.anl.gov

based on slides from

Ben Clifford benc@ci.uchicago.edu

Scott Koranda skoranda@uwm.edu