LIGO Data Grid: Making it Go

Scott Koranda

University of Wisconsin-Milwaukee

Grid Middleware & the Gap

LIGO Data Grid leverages the *best* of grid middleware

- Globus Toolkit
  - » GSI and X.509 certificates for authentication/authorization
  - » GridFTP for moving data and files
  - » Replica Location Service (RLS) for data discovery
- Condor
  - » Condor for high throughput computing on clusters
  - » Condor DAGman for workflow execution
- Virtual Data Toolkit (VDT)
  - » pyGlobus
  - » Pegasus
  - » many, many more

Grid Middleware & the Gap

Using these middleware building blocks LIGO has...

- developed tools for data discovery
- developed tools for managing complex workflows

...but the building blocks are middleware

Alone the middleware does *not* put tools into the hands of LIGO scientists

An important GAP is left between the middleware and what LIGO scientists need to work efficiently

Grid Middleware & the Gap

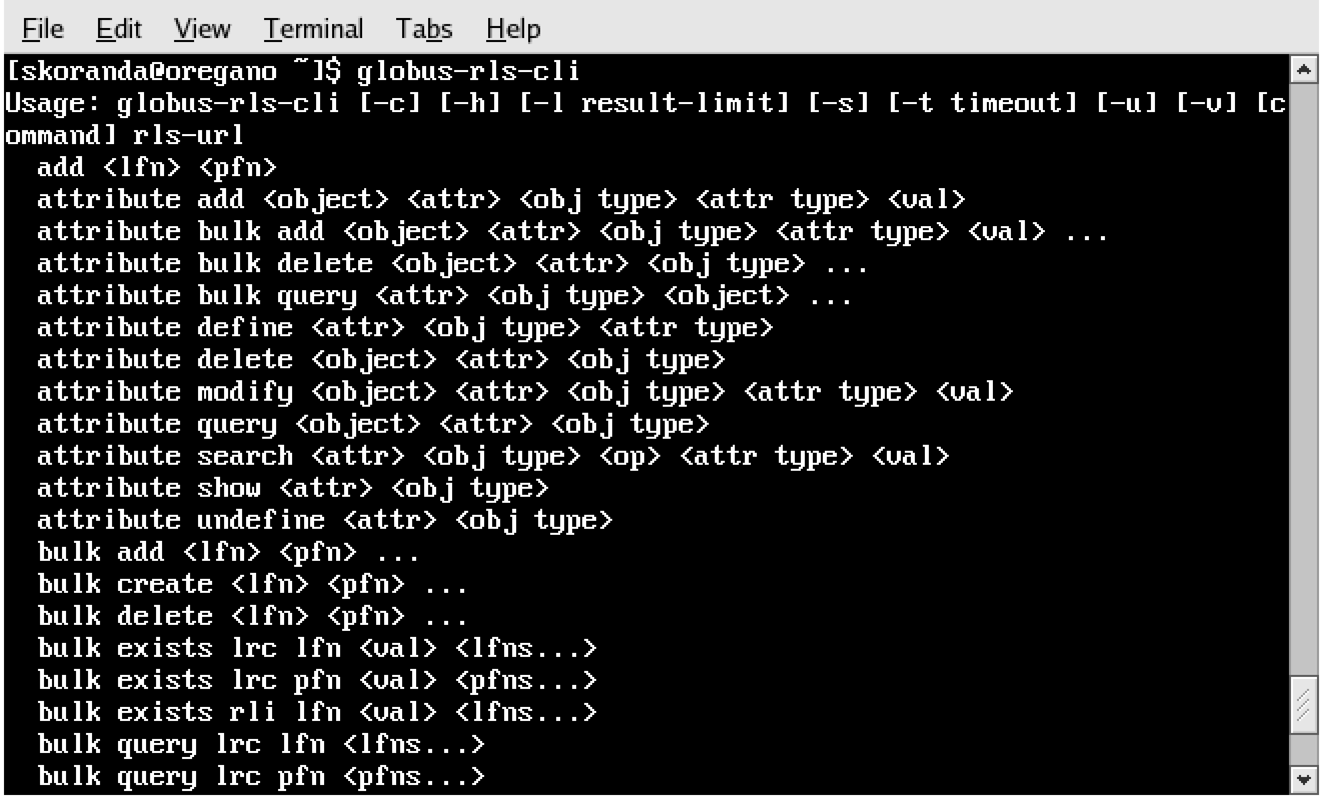
Different science projects (LIGO, SDSS, CMS, ATLAS) can leverage much of the same grid middleware...

...yet they are distinct experiments and so there will *always* be gaps between the common middleware and tools LIGO scientists can use

Data discovery a good example:

- middleware tool is Globus Replica Location Service (RLS)
- GT provides the "simple" command line tool globus-rls-cli

Grid Middleware & the Gap



```
[skoranda@oregano ~]$ globus-rls-cli
Usage: globus-rls-cli [-c] [-h] [-l result-limit] [-s] [-t timeout] [-u] [-v] [c
ommand] rls-url
  add <lfn> <pfn>
  attribute add <object> <attr> <obj type> <attr type> <val>
  attribute bulk add <object> <attr> <obj type> <attr type> <val> ...
  attribute bulk delete <object> <attr> <obj type> ...
  attribute bulk query <attr> <obj type> <object> ...
  attribute define <attr> <obj type> <attr type>
  attribute delete <object> <attr> <obj type>
  attribute modify <object> <attr> <obj type> <attr type> <val>
  attribute query <object> <attr> <obj type>
  attribute search <attr> <obj type> <op> <attr type> <val>
  attribute show <attr> <obj type>
  attribute undefine <attr> <obj type>
  bulk add <lfn> <pfn> ...
  bulk create <lfn> <pfn> ...
  bulk delete <lfn> <pfn> ...
  bulk exists lrc lfn <val> <lfns...>
  bulk exists lrc pfn <val> <pfns...>
  bulk exists rli lfn <val> <lfns...>
  bulk query lrc lfn <lfns...>
  bulk query lrc pfn <pfns...>
```

Filling the Gap

Filling this gap requires *dedicated LIGO effort*

- *work with LIGO scientists to understand use case*
  - » *how can I find data from a particular site for a particular interval of time in a particular format?*
- *understand the underlying middleware (RLS)*
  - » *how does the API work, can we use LSC favorite glue languages, what is necessary for deployment on LSC computing resources?*
- *write the code...*
- *test the code...*
- *package the code...*
- *document the tool...*
- *deploy the tool...*
- *support the users!*

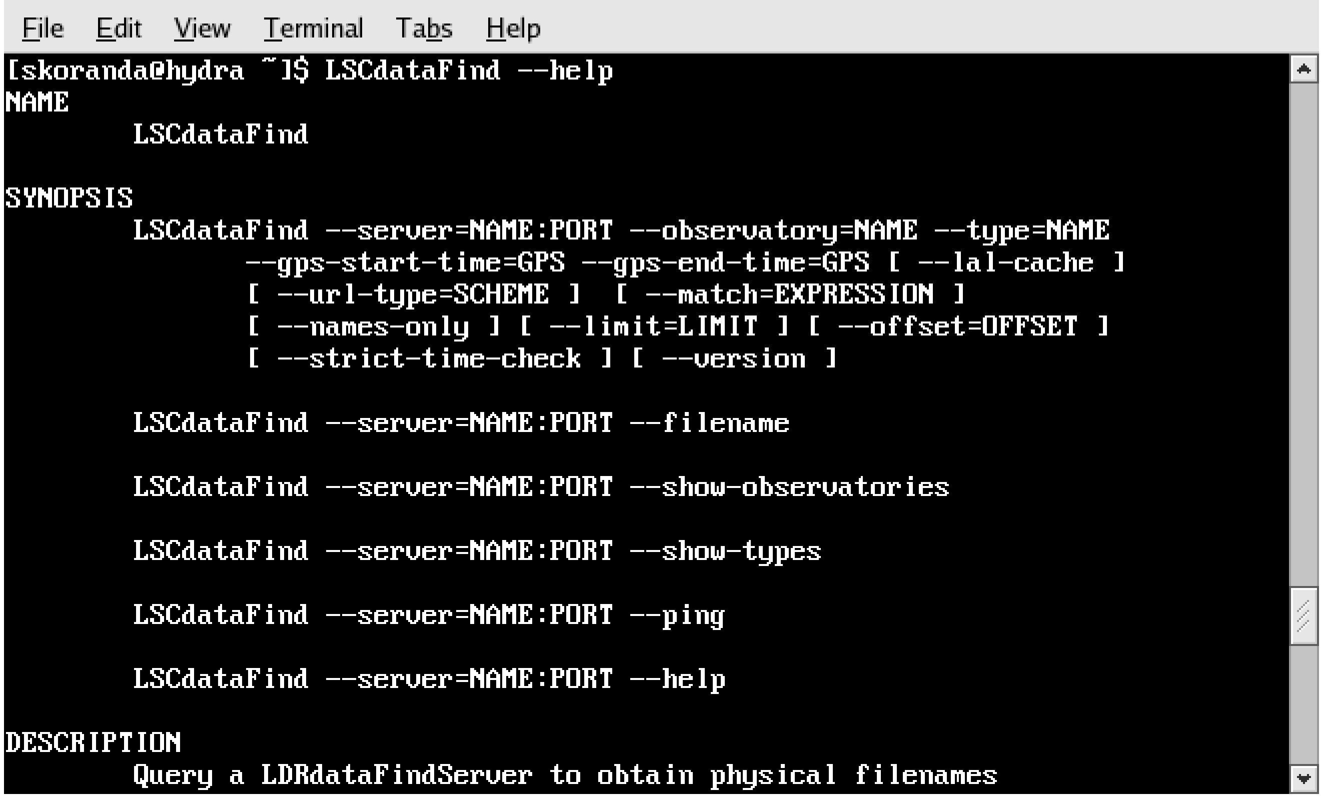
Filling the Gap

Bridging these gaps is not “set and forget”

- Tools must continually evolve as LIGO data analysis matures
- LIGO computing professionals must constantly incorporate scientist feedback, evolve, and release again

Filling the Gap

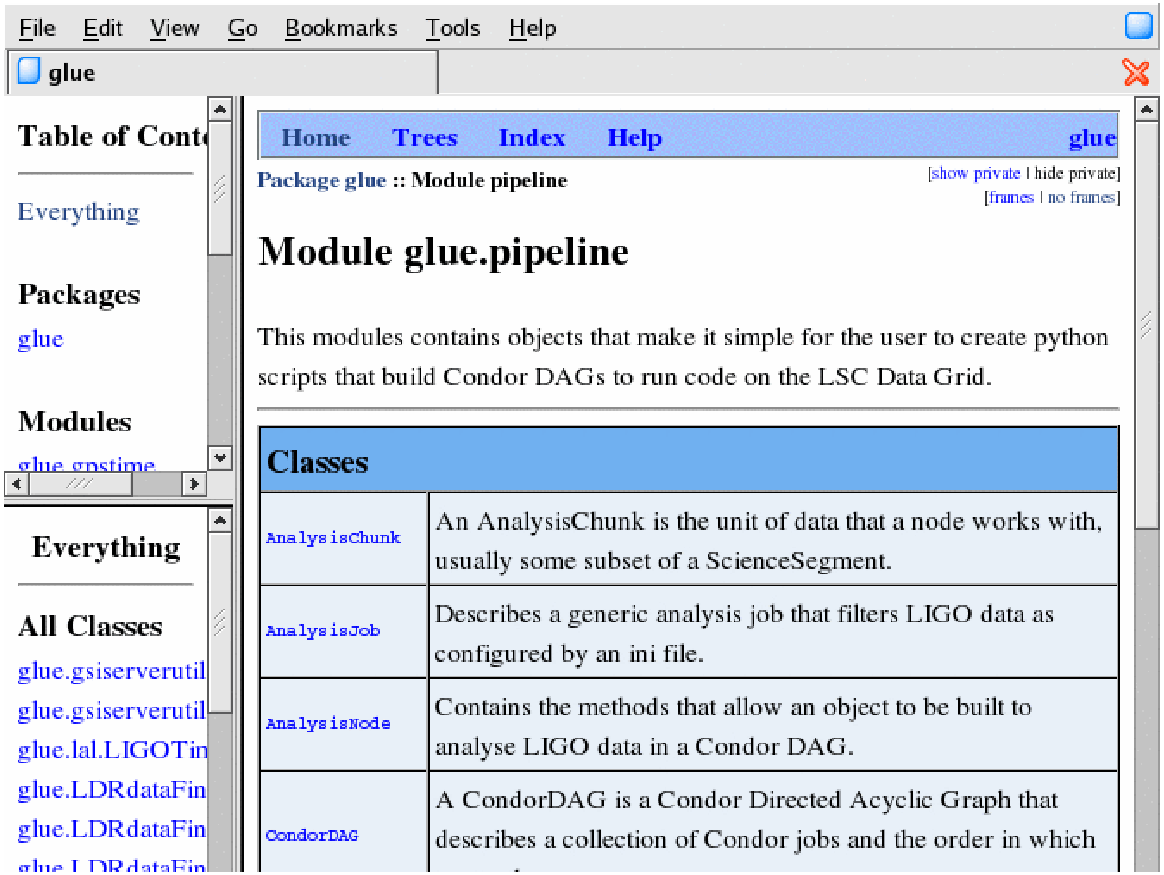
Data discovery in LIGO using LSCdataFind



```
[skoranda@hydra ~]$ LSCdataFind --help
NAME
        LSCdataFind

SYNOPSIS
        LSCdataFind --server=NAME:PORT --observatory=NAME --type=NAME
               --gps-start-time=GPS --gps-end-time=GPS [ --lal-cache ]
               [ --url-type=SCHEME ]  [ --match=EXPRESSION ]
               [ --names-only ] [ --limit=LIMIT ] [ --offset=OFFSET ]
               [ --strict-time-check ] [ --version ]

        LSCdataFind --server=NAME:PORT --filename

        LSCdataFind --server=NAME:PORT --show-observatories

        LSCdataFind --server=NAME:PORT --show-types

        LSCdataFind --server=NAME:PORT --ping

        LSCdataFind --server=NAME:PORT --help

DESCRIPTION
        Query a LDRdataFindServer to obtain physical filenames
```

Filling the Gap

Tools for constructing LIGO workflows (built on top of Condor DAGman):



Package glue :: Module pipeline

Module glue.pipeline

This modules contains objects that make it simple for the user to create python scripts that build Condor DAGs to run code on the LSC Data Grid.

Classes	
AnalysisChunk	An AnalysisChunk is the unit of data that a node works with, usually some subset of a ScienceSegment.
AnalysisJob	Describes a generic analysis job that filters LIGO data as configured by an ini file.
AnalysisNode	Contains the methods that allow an object to be built to analyse LIGO data in a Condor DAG.
CondorDAG	A CondorDAG is a Condor Directed Acyclic Graph that describes a collection of Condor jobs and the order in which

Bigger Gaps in Middleware

For some important LIGO tasks the grid middleware stack comes up short:

- TclGlobus
  - » many useful LIGO tools pre-date LDG written using Tcl/Tk
  - » fully incorporating these tools into LDG requires a TclGlobus
- LIGO Metadata Service
  - » metadata for different science experiments is mostly unique
  - » little investigation so far in larger grid community
  - » LIGO has working prototype that (just barely) scales for S5

Prototyping, developing, and deploying grid middleware from within LIGO is sometimes necessary...but comes with a cost

Maturing the LIGO Data Grid

- Metadata service
  - » LIGO specific (for at least 3 years) requires prototyping, developing, deploying, support
- Resource discovery service
  - » requires wrapping tools like Globus MDS or Condor ClassAds and developing LIGO specific schema
- Monitoring services
  - » requires investigation and sampling of a large number of possible solutions, then customization, deployment, and documentation
- Advanced workflow planning
  - » requires integration of Globus Pegasus and VDS into LIGO GLUE toolset
- Brokering services
  - » still active research area...requires LIGO use case investigation and feedback into grid middleware community

Maturing the LIGO Data Grid

- Authentication & Authorization (LIGO CA)
  - » deployment underway, requires documentation and solid long term user support
- VO management
  - » initial LIGO deployment ongoing, requires continued investigation of the VO privilege project led by US CMS & ATLAS
- LDAS
  - » requires tight integration into LDG authorization model
- LDR
  - » requires metadata scaling solutions, advanced replica services to be developed and deployed
- each of these requires filling gaps between the common middleware and LIGO scientists with wrapping, deploying, documentation, support...

Maintaining the LDG Fabric

None of the LIGO Data Grid services will run without hardware & networking...

...ongoing administration of all LDG computing resources is *critical*...

...as is administration of LDG services at each LDG site and dedication to local support

LIGO Evolving the LIGO Data Grid LSC

LIGO Data Grid exists now and is enabling LIGO science...

...but realizing the full potential of the LIGO Data Grid and exposing the full science potential of LIGO requires a constant investment in LIGO computing professionals around the LIGO Data Grid to develop, deploy, document, and support the services and tools LIGO scientists need.