



The QCDgrid Software Suite

An Introduction

Daragh Byrne
Applications Consultant, EPCC
d.byrne@epcc.ed.ac.uk
+44 131 651 3395

- Overview of QCDgrid system
 - Main components of the QCDgrid software suite
 - Data grid
 - Metadata catalogue
 - Job submission system
- Operation
 - Client tools
 - How the back end works
 - Security
- Practical demonstration

- QCDOC coming on-line
 - Large amount of data will be produced
- This data needs to be managed
 - Stored
 - Accessed
 - Shared
 - Searched for
 - Moved around
 - Processed

- Three main components
- QCDgrid is primarily a **data grid**
 - Store data online (disk)
 - In this case, the result of QCDOC work
 - Reliability through redundancy
 - Multiple copies of files in geographically distinct locations
- QCDgrid offers a metadata catalogue service (MDC)
 - Configurations/ensembles are described using QCDml1.1
 - Describes various (physics and otherwise) parameters used in simulations
 - Allows users to search for data matching these parameters – can find useful data
- Also a job submission grid
 - Primarily for post-processing jobs on clusters

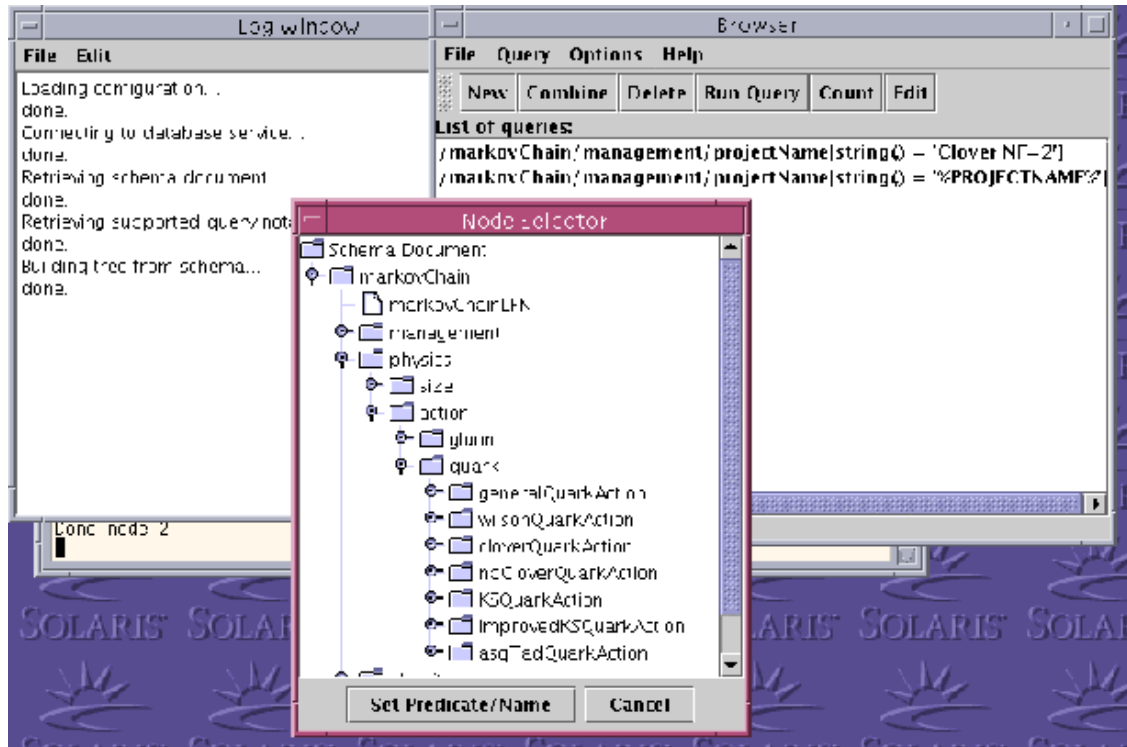
- Store binary data files
 - Gauge configurations/ensembles (could be used to store any data)
 - Eventually store large numbers of these files
- Store files:
 - Reliably
 - Transparently
 - Files have a logical filename – that is all you need to know to access a file
 - Hides details of physical storage
 - Securely
- Retrieve files

- Allows you to search for particular physics data
 - Search by physics parameters, creation data, who created by...
 - Can help avoid replication of work by allowing you to search first
- Graphical client provides easy way to search catalogue

- Allows submission of jobs to specified machines
 - E.g. NGS machines for post-processing of gauge configurations

- `put-file-on-qcdgrid <file-or-directory> <logical-file-name>`
 - Used to place data files on the grid
 - Must use `-R/--recursive` to put directories of files on the grid
- `get-file-from-qcdgrid <logical-file-name-or-directory>`
 - Used to copy files from grid to local directory
 - Must use `-R/--recursive` to get directories of files from the grid
- `qcdgrid-delete <logical-file-name-or-directory>`
 - Can only be used by grid administrators!
- Need to be configured by `nodes.conf` and `nodeprefs.conf`
- We'll have a look at these in action later

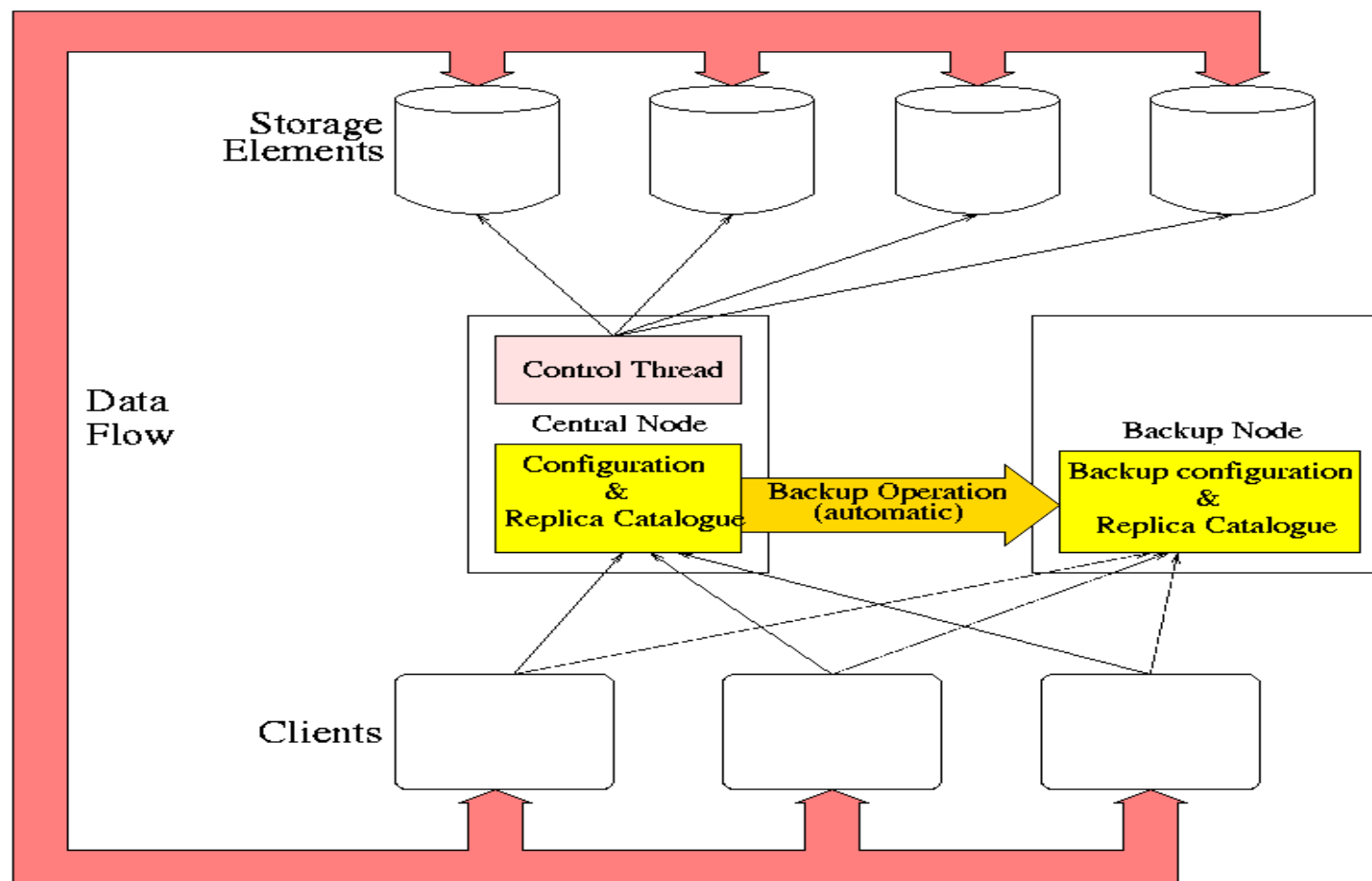
- Java programme
- Graphical interface that allows you to define searches for data and retrieve results
 - Automatically updates UI when the QCDML schema changes



- `qcdgrid-job-submit` executable
- Simple wrapper over Globus GRAM
- Designed for post-processing jobs
- Will pull grid data automatically to the job site

- Tier 1 system
 - 50TByte SAN system
 - QCDOC front end directly connected
 - Hold one copy of all UKQCD data
- Tier 2 systems
 - Edinburgh, Liverpool, Southampton, Swansea
 - Linux server + RAID array disk
 - 12.5 Tb per system
- 50 TByte capacity + 50 TByte replicated

- The system consists of three types of grid node:
 - Central (control) node
 - Backup node
 - Storage element node
- Central node controls system
 - Keeps track of file locations
- Backup node
 - In the event of failure of central node, allows read access to data grid
- Storage nodes
 - Provide storage space on local filesystem



- Coordinate data storage
 - Maps logical file names to physical locations
 - Ensures duplication of files i.e. two (or more) copies of each exists at all times
 - Ensures consistency of data on the grid
 - i.e. verifies no corruption has occurred
- Keep track of storage nodes
 - Keep track of available space on each node
 - Manage disabled nodes (temporarily offline storage nodes)
 - Manage node retirement (permanently removed storage nodes)

- Uses Globus Replica Location Service (RLS) to map logical filenames (LFNs) to physical locations
 - Obviously Globus must be installed
- Connects to locally running RLS
- In the production QCDgrid system, the central node is `edqcdgrid.epcc.ed.ac.uk`
- Responds to messages from client tools and other nodes, both synchronously and asynchronously

- Store the data on disk
- Each has Globus installed
- Each has the QCDgrid software installed
- Each have a storage area (a logical directory)
 - Has a `NEW` subdirectory for files that are uploaded (inbox)
- Most operate using a dedicated `qcdgrid` account

- Keeps a current copy of the replica catalogue on the central node
 - Replication is automatic and periodic
- Runs continuously
- Read-only
 - For data consistency
- Clients will contact this node in the event the central node is down

- Stored in eXist XML database
- eXist runs as a web application in the Tomcat application server
- Search for interesting metadata -> get pointer to the data file's LFN

- **Submit file to grid**
 - `put-file-on-qcdgrid` uses `gridftp` to get the amount of free space on each storage node
 - Selects suitable node
 - Gridftps the file to that node
 - Sends message to control node
 - Replication will take place under via the control node
- **Get file from grid**
 - `get-file-from-qcdgrid` contacts central node to find out where copies of a file are stored
 - Selects a node and gridftps file
- **Metadata browser**
 - Creates XPath queries based on graphical query builder
 - Interacts directly with eXist server

- Based on the Globus Security Infrastructure (GSI)
 - Uses digital certificates issued by UK eScience Authority
- Obtain your certificate by following the instructions at <http://ca.grid-support.ac.uk/>
 - Follow the “Request a Certificate” link

- Please use our bug tracking software on our NeSCForge site http://forge.nesc.ac.uk/tracker/?group_id=13
- Allows us to keep track of queries and solutions
- Can email general queries to qcdgrid@epcc.ed.ac.uk

- <http://www.epcc.ed.ac.uk/~daragh/qcdgrid/tutorial.html>