DIRAC and Rucio

Janusz Martyniak and Simon Fayer

Imperial College London

GridPP
UK Computing for Particle Physics

What is RUCIO?

Rucio is an open source framework which provides functionality to store and distribute scientific data. Data distribution in Rucio is rule-based which differs from the approach used by DIRAC.

A file is the smallest object in Rucio and it is mapped to a file in DIRAC. Files can be grouped into datasets, which in turn can be grouped into containers.

Directories with files and directories -> datasets in Rucio
Directories containing directories - > containers in Rucio

Files, datasets and containers follow the same naming scheme which is made up of 2 strings: a scope and a name. This combination is called a Data Identifier, a DID.

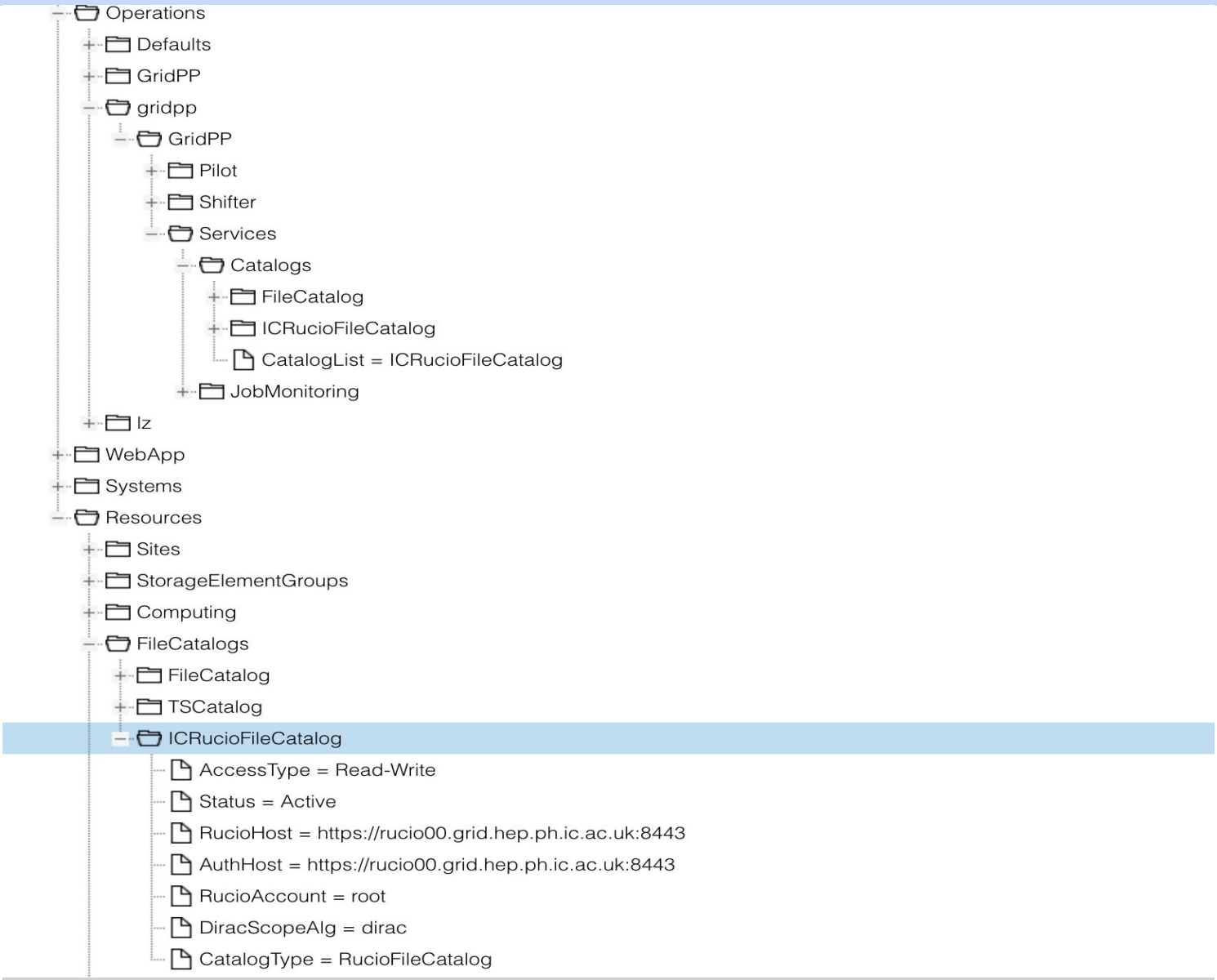
Enabling Multi VO Rucio in Multi VO DIRAC

- The GridPP implementation is based on the BELLE II single VO Rucio plugin.
- The original BELLE II implementation was covered in detail by Cedric in his talk at the previous DIRAC workshop.
  https://docs.google.com/presentation/d/143o0eD8CiDuWzLVIlReFlibvbxkT1kowjvGObcWp3SI/edit#slide=id.p
- The GridPP implementation concentrates on adding multi VO functionality to work with a Multi VO Rucio server developed at RAL.
- We try to keep backward compatibility with existing BELLE II code.
- The implementation consists of a Rucio File catalog client, users, scopes and SE synchronisation agent and the RSS sync agent. All 3 required modifications to serve multiple VOs. The Rucio server code was also modified.

Rucio client configuration

- Rucio client would normally use a config file which stores necessary parameters. We opted for a system which would work w/o this file and store all “static” parameters (e.g Rucio server location) in DIRAC CS and pass them to Python client along with user credentials available in DIRAC
- This requires a few changes in Rucio server, so the file becomes optional.
- BELLE II uses a config file, so we check this first, if it exists, parameters stored there take precedence.

Example of DIRAC configuration



Mapping the Rucio DIDs to DFC LFNs

Scope is a Rucio concept, which allows to:

- divide the namespace between VOs, users, activities
- it is used for replication policies and accounting

It is unknown to DIRAC, so in the *DIRAC scope algorithm* we decided to put into the LFN as the second element, after the VO name, for example:

	Scope	Name
RUCIO DID:	*user.john.smith*	*:/some/path/to/file.root*

DFC LFN: */gridpp/user.john.smith/some/path/to/file.root*

(the scope for user john.smith would be automatically created by the sync agent, covered later)

Scope algorithm and a (lack of) a config file…

At the moment scope extraction algorithms are hardcoded in Rucio. The config file (on the server and client side!) is used to pick the required algorithm. The algorithms selected on the client and server sides have to match.

We added a DIRAC scope extraction algorithm which is selected when no client-side config file is present.

The config file only allows a *single* algorithm to be active on the server. There is work going on in the UK to extend Rucio policy packages to scopes.

File Catalog methods implementations

BELLE II implemented most of the FC client methods. The read-only methods required no modifications for multi-VO

The *addFile()* method was more tricky, since it delegates all the catalog related work to the server.

There were multiple changes need in the Rucio server code (in flaskapi, API and core layers) to accommodate multi-VO handling.

Rucio Agents

RucioSynchronizerAgent creates user accounts in Rucio and scopes for them reading information from DIRAC CS. The agent was modified to create Rucio users and scopes only for the VOs configured to use Rucio. The agent also creates RSEs in Rucio based on the info in DIRAC CS.

RucioRSSAgent sets the RSE status according to the SE status published by the RSS:

1. active, degraded -> available in Rucio
2. banned -> disabled in Rucio

Synchronisation is only done for eligible VOs.

Conclusions

The Rucio code modifications have been accepted and released. DIRAC RucioFileCatalog plugin is available in DIRAC v7r3. The system still needs to be tested on a production Rucio server at RAL.

Future work: BELLE II are currently working on DIRAC metadata handling in Rucio. Once ready, we will add it to vanilla DIRAC.

Many thanks to my colleagues for help and BELLE II for assistance when debugging the code.