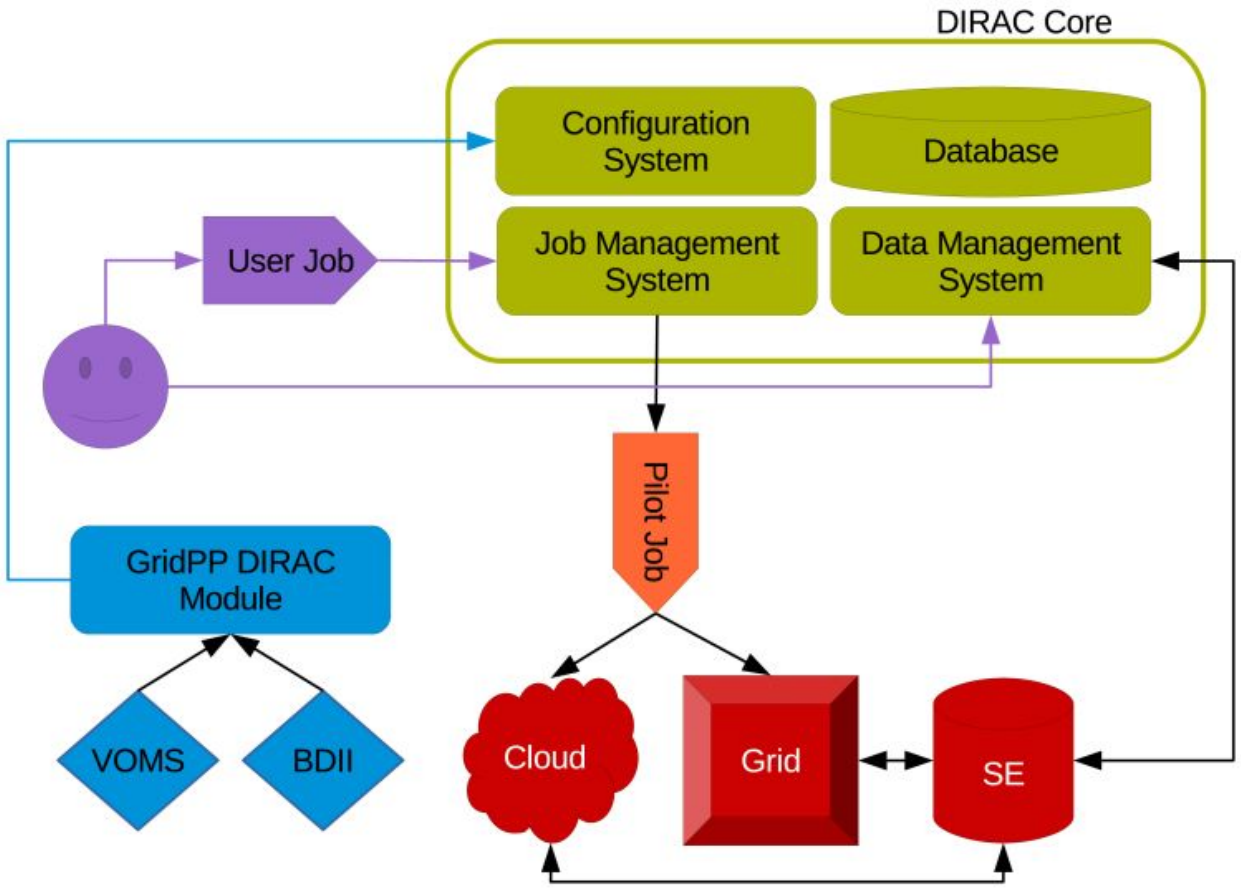
Imperial College Digital Asset Status: DIRAC

Janusz Martyniak, Simon Fayer & Daniela Bauer

Reminder: What is DIRAC (not DiRAC)

- DIRAC is a workload management and data management system
  - It is maintained by the DIRAC consortium as an Open Source project
- GridPP run a DIRAC instance for non-LHC communities in the UK



Overview

Planned deliverables schedule at start of project. Deliverables are independent of each other.

	Deliverable/Month	1-6	7-12	13-18	19-24
1	DIRAC Metadata (Transformation System)	Janusz/Alex			
2	DIRAC Resource Status System		Janusz/Alex		
3	DIRAC RUCIO			Janusz/Alex	
4	DIRAC IRIS Cloud				Daniela & Simon

Current status

	Deliverable/Month	1-6	7-12	13-18	19-24
1	DIRAC Metadata (Transformation System)	Janusz/Alex			
2	DIRAC Resource Status System		Janusz/Alex		
3	DIRAC RUCIO			Janusz/Alex	
4	DIRAC IRIS Cloud			Daniela & Simon	

Deliverable 4: Enhance direct cloud submission in multi-VO DIRAC

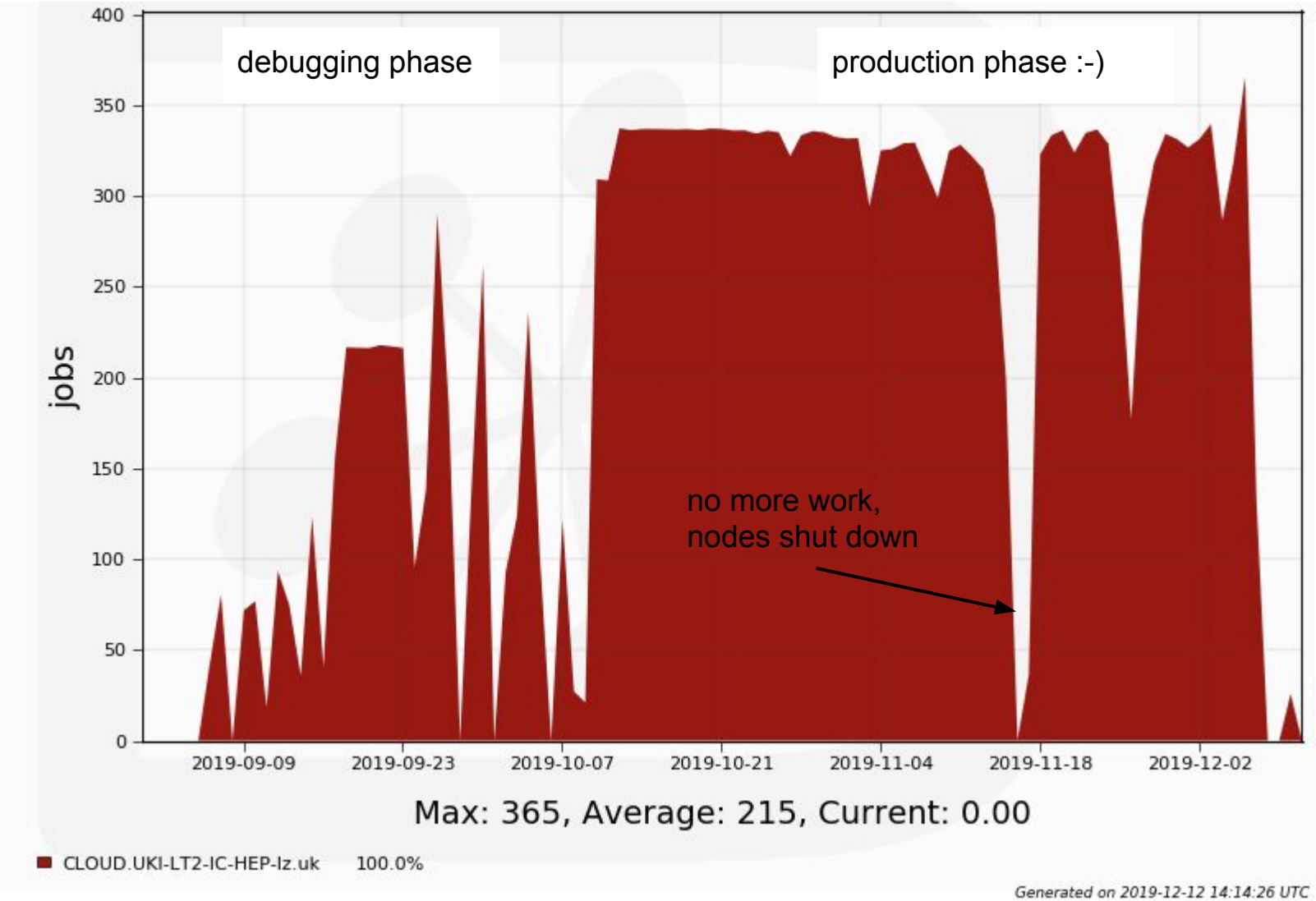
DIRAC provides an extension (“VMDIRAC”) for direct cloud submission.

- Initially developed against OpenNebula with only basic OpenStack support
- Implemented correct handling of application credentials
- Fixes/enhancements after large scale testing
- Improved VM runtime contextualisation by switching from ad-hoc scripts to cloud-init.
  - This reduced the contextualisation failure rate and start-up overhead.
  - Uses singularity to isolate the payload from the pilot credentials, improving security.

Todo:

- Container support for users
- Add more per community configuration options

GridPP DIRAC to Imperial IRIS Cloud



Deliverable 1: DIRAC Multi-VO File Catalogue Metadata System

Current FC metadata implementation is not VO-aware. Users can attach metadata information to files and directories but can also modify/delete each info across VOs.

The aim is to separate metadata belonging to different VOs in such a way that it is transparent to a user.

- No changes at a client side (no need to update the DIRAC UI). A user stores and retrieves metadata information as before. The VO information is added/removed in flight.
- This feature is supplied as an optional *plugin.*
- This requires updates to both the DIRAC server and catalogue.

DIRAC Multi-VO Metadata implementation status

DIRAC uses *indexed* metadata tables optimised for access speed. It is implemented in such a way that separate tables are used for every metadata name.

In order to maintain this approach the multi VO catalogue is implemented by adding a VO suffix to relevant table names.

The code has been submitted, reviewed and accepted by the DIRAC project.

DIRAC project are planning to integrate multi-VO setup in their automated testing:

- Need to review test results once available.

Deliverable 2: DIRAC Resource Status System (RSS)

RSS automatically disables faulty resources from receiving users' jobs.

1. RSS stores status of the resources in DIRAC (Active, Bad, Banned, Probing)
2. The status can be assigned by an admin (via dirac-rss-* commands) manually
3. Can also be determined by the RSS by "intelligent" monitoring
   a. The core is a generic policy system
   b. Which can be used for monitoring and management
   c. Change the status of a resource automatically
4. RSS can handle status of sites and resources (CE, SE, FTS)
5. Policies could:
   a. Contact GOCDB to get the status of a site or resource
   b. Actively test a resource by examining a pilot submission success ratio, if this is not satisfactory, then take an *action.* This could mean changing a status of a resource or issuing a notification.
6. Policies and their actions are highly configurable in the DIRAC configuration

RSS and multiple VOs

The RSS is a currently single VO system. The aim of the project is to make the RSS multi-VO compatible.

Multi VO modifications require:

1. Multiple tables in the DB schema require adding a VO column.
2. dirac-rss-* client-side commands have to be modified.
3. Policy evaluation chain has to be made VO-aware
4. Job submission has to take into account resource status on a VO by VO basis.

Multi-VO RSS Status

1. Database schema modified
2. Test Dirac instance configured to use multi-VO policies. We are currently focusing on two core policies:
   a. Site/Resource status from GOCDB configured: note that GOCDB does not support VO dependent information
   b. We evaluate pilot success efficiency on a VO by VO basis:
      i. Small adjustment to the existing pilot policy were necessary
      ii. the code to fill in a pilot efficiency cache has been re-implemented for multiple VOs,
      iii. Policies based on pilot efficiency are evaluated
   c. a) and b) combined (as a usage example, not a strong rule) and stored in the RSS
3. Still to be done: user side VO handling (dirac-rss-*)
4. Modify the DIRAC pilot submitter to obey the VO-aware resource status.

Deliverable 3: DIRAC-Rucio integration

The aim is to use DIRAC for workload management and Rucio for data management.

The integration is built on these existing DIRAC components:

1. Use user/group/VO registry
2. Use storage elements as defined in DIRAC
3. Drop DIRAC File catalog for VOs wanting to switch to Rucio.

Preliminary work has started on this, the main project is planned start in Jan 2020.

Status: Test Implementation at Imperial

1. Configure a VO to use a Rucio Catalog plug-in
2. Use DIRAC data management client API or scripts to contact DIRAC
3. DIRAC selects a correct plugin based on VO information stored in the proxy
4. Plug-in performs required data handling operations using Rucio API or a REST interface.
5. User identity is being passed in to Rucio

Next step depends on RAL multi-VO Rucio server, to be provided as part of RAL Rucio digital asset.

Summary

All parts of our Digital Assets are progressing and are roughly on track with the original project plan.