

GridPP

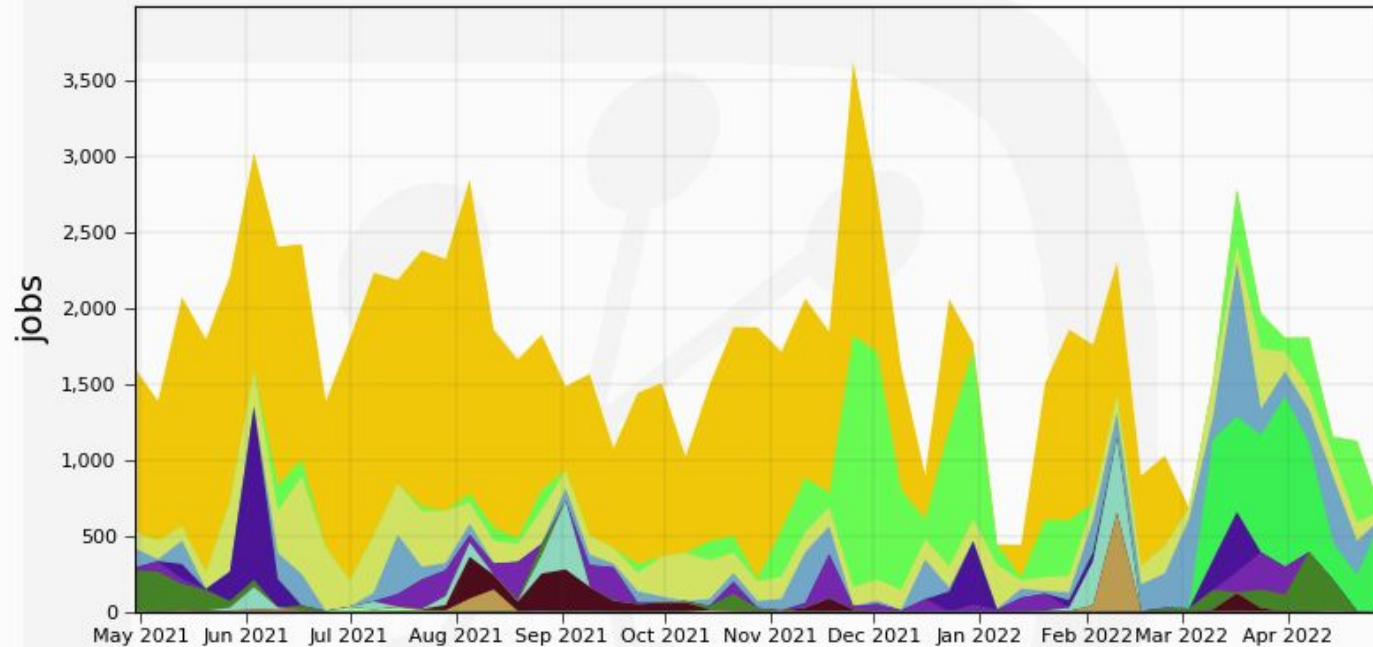
Daniela Bauer, Simon Fayer & Janusz Martyniak

Overview

- The GridPP DIRAC is a **multi-VO** instance. It is configured for 18 VOs, of which typically ~9 are using it actively.
- The supported communities are particle physics/neutrino/astrophysics projects, though this is not a condition for support.
- All communities use DIRAC as a workload management tool.
- The majority also use the data management facilities.

Overview

Running jobs by UserGroup
52 Weeks from Week 17 of 2021 to Week 17 of 2022

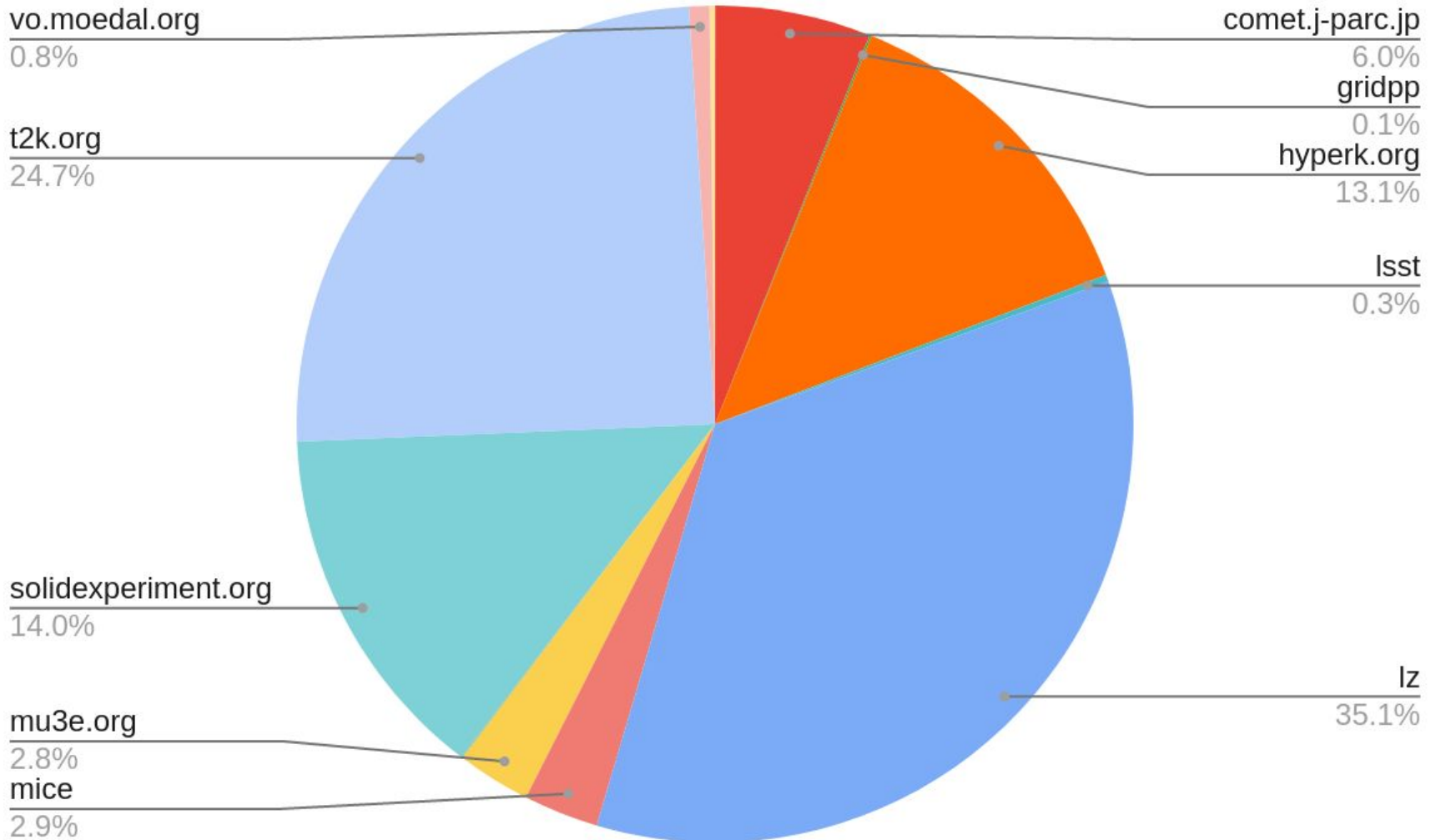


Max: 3,615, Min: 437, Average: 1,747, Current: 669

na62.vo.gridpp.ac.uk_user	52.3%	hyperk.org_user	1.8%
vo.moedal.org_user	11.0%	lz_user	1.1%
snoplus.snolab.ca_production	10.8%	mice_user	0.1%
lz_production	7.1%	gridpp_user	0.0%
na62.vo.gridpp.ac.uk_production	5.5%	skatelescope.eu_user	0.0%
solidexperiment.org_production	3.1%	snoplus.snolab.ca_user	0.0%
t2k.org_user	2.9%	vo.northgrid.ac.uk_user	0.0%
pheno_user	2.4%	hyperk.org_production	0.0%
mu3e.org_user	1.9%	... plus 4 more	

Generated on 2022-05-06 11:37:42 UTC

DFC: ~30 mio files



Stakeholders

- GridPP (Grid computing for particle physics):
<https://www.gridpp.ac.uk/>
- Imperial College (HEP group):
<https://www.imperial.ac.uk/high-energy-physics>
- IRIS (Digital research infrastructure for eScience):
<https://www.iris.ac.uk/>
- SWIFT-HEP (SoftWare and Infrastructure Technology for High Energy Physics): <http://swift.hep.ac.uk/>

GridPP and Imperial College fund personnel to run the service; GridPP also contributes the DIRAC hardware.

IRIS and SWIFT-HEP grants are for project work only: Typically 6-18 months long, must have a finished product (read: code) at the end.

Priorities

- Our first priority is to keep the **production server** running and in good shape. Main tool: Nagios monitoring.
- **Updates:** Production server should run a recent version of the base branch (currently v7r2). Main tool: Designated pre-prod server. Tests functionality and bug fixes.
- **Long-term upgrades:** We typically start 6 month to a year in advance for major production upgrades to make sure the new release contains all the functionality we need. We are currently **working on** v7r3/python3. Main tool: Pre-prod server.
- **Project work:** Feature development. Can be user or admin driven. Main tool: Janusz' test server (now v7r3).

Technical nitty gritty

- Which DIRAC version do you use in production? Have you migrated to Python3 (client/pilot/server)?
 - Production: v7r2p40 (all python2, apart from (optional) UI)
 - Pre-prod/Devel: v7r3 (everything python3)
- What do you use DIRAC for, and which DIRAC functionalities you don't use, and why?
 - Use: Workload management, data management
 - Don't use: Transformation/Workflow management: Mainly due to lack to time to commission/maintain it. But working on it.
- Do you have a DIRAC extension? Why? (If yes, do you think some of it could become part of the vanilla projects?):
 - Autoconfig tool that configures our sites just how we like it.
 - Not suitable for anyone else, unless you really like our naming scheme/queue setup.

Technical nitty gritty continued.....

- What is your biggest frustration with DIRAC?
 - Susceptibility to upstream changes (conda, CI)
 - Dependencies on external services (message queues) etc. While we can see it might be necessary, we don't have CERN to run these things for us.
 - Can you please stop messing with the `dirac.cfg/pilot.cfg`.
 - Everything should be versioned.
 - Better error handling. There is progress, but we have yet to debug a problem without having to insert copious amounts of print statements.
- You can magically add one feature to DIRAC, what is it?
 - See frustrations.
 - Rucio like data placement rules.

... and more

- Any notable operations incident in the last year?
 - Person most likely to break GridPP DIRAC: Daniela ('creative' attempts at bug fixing, git mishaps).
 - Person most likely to fix GridPP DIRAC: Simon (pythonic fixes, git mastery).
 - Our users are incredibly well behaved.
 - Our weak point is the data catalogue which can get overloaded:
 - i. We will probably try to mitigate this by splitting it into multiple instances.
- How would you rate the communication?
 - Communication generally works.
 - i. It's helpful to have the github issues and discussions on a watch list. (Though we still don't see what's wrong with a plain old google group :-)
 - ii. BiLd meetings, while (too?) long, are useful.
 - iii. Ditto hackathons.

Projects: CloudComputingElement

- Background:
 - GridPP has a strong interest in using cloud resources via DIRAC.
 - VMDIRAC was a separate project to manage cloud resources in DIRAC.
 - In v7r3 the VMDIRAC repo was merged with DIRAC repo, to ease maintenance and installation.
 - v7r3 CloudDirector still contains a lot of similar logic to the SiteDirector:
Unnecessary overhead maintaining both.
- In v7r3 we introduced the CloudComputingElement which can be used by the SiteDirector.
- The CloudComputingElement conceptually treats a VM as a pilot job (using the instance id as pilot job reference).
- Documentation:
<https://dirac.readthedocs.io/en/integration/CodeDocumentation/Resources/Computing/CloudComputingElement.html>

CloudComputingElement continued

- Uses apache-libcloud (<https://libcloud.apache.org/>) – “A Python library for interacting with many of the popular cloud service providers using a unified API” underneath.
 - We submitted on patch (to be able to use app credentials) to libcloud beforehand.
- Python3 only as it depends on a python3 only version of apache-libcloud ($\geq 3.4.0$).
- We are keen to work with other DIRAC users to make their interfaces work. (Yes, we are offering to write code for you.) Please talk to us.
- Known issues:
 - Token support not tested, but “should work”.
 - OpenNebula interface in apache libcloud potentially out of date.

Projects: Pilot logging

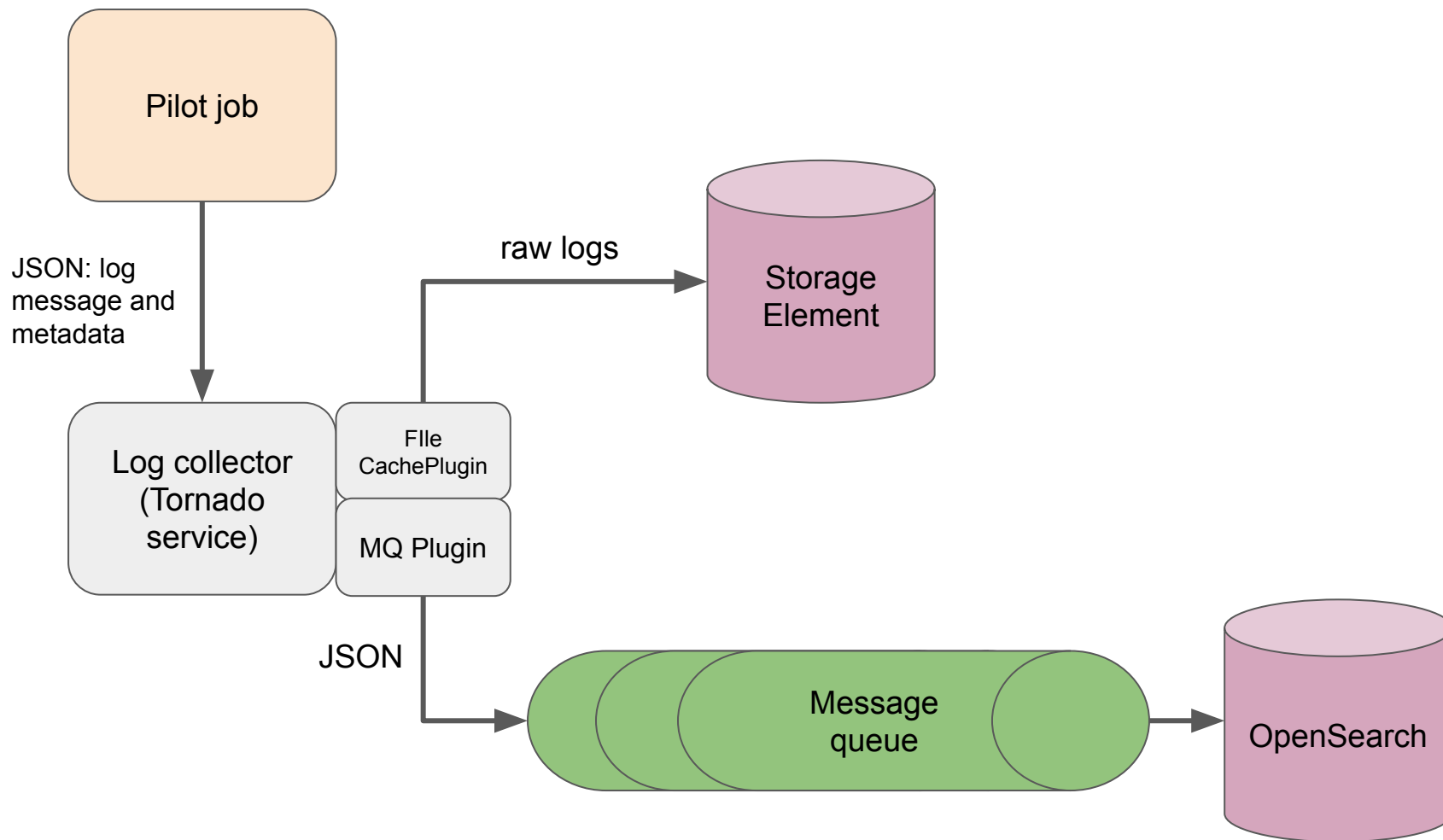
DIRAC sends the pilot job output to *stdout* and *stderr*:

- Traditionally collected on CE, but only available after job finishes (or not at all in some failure cases).
- Not available at all on some resources, such as clouds.
- Hard to collect and analyse systematically.

New pilot logger under development:

- Send batches of log message to a central DIRAC service at run time.
- Consistent across all compute resources.
- Can be analysed systematically with e.g. OpenSearch.

Pilot logger overview



Pilot Logging Status

- Changes to pilot wrapper and code:
 - Remove existing remote logger code and its dependencies
 - Implement a new logging handler to send logs in batches
- File cache plugin: implemented.
- Message Queue plugin: skeleton exists.
- This code is under active development.
 - If you want to participate in development or have a use case that is not covered, this is usually discussed in the development (“BILD”) meetings.

Questions ?

The GridPP DIRAC team can be reached under:

lcg-site-admin@imperial.ac.uk

If you prefer you can also tag us on github:

<https://github.com/DIRACGrid/DIRAC/discussions>

@sfayer @marianne013

pilot logging and rucio: *@martynia*

We are also usually in the developers meeting(*) if you prefer just to talk to us.

(*) Ask Federico if you would like to receive notifications about this.