DIRAC
Current, upcoming and planned capabilities and technologies
CHEP 2023

May 9\textsuperscript{th} 2023

Alexandre F. Boyer
alexandre.boyer@cern.ch
Federico Stagni
federico.stagni@cern.ch

European Organization for Nuclear Research
Meyrin, Switzerland
Introduction

Have access to remote computing/storage resources but don't know how to use them?

There exist many paradigms and implementations to interact with shared computing and storage resources.

- Batch systems (BS), Computing Elements (CE), Storage Elements (SE), communication protocols...

DIRAC aims at providing an abstraction layer between different user communities and these different resources.
DIRAC
An open source middleware for distributed computing

- Started as an LHCb project.
- Experiment-agnostic since 2009.
- Developed by communities, for communities.
  - GitHub hosted
  - Publicly documented, active assistance forum, yearly users workshops, open developers meetings and hackathons.
A framework shared by multiple experiments/projects, both inside HEP, astronomy, and life science: Experiment agnostic, Extensible, Flexible.
Getting the big picture

A few presentations/posters about DIRAC:

- Standardizing DIRAC’s Cloud Interfaces
- Improved Pilot Logging in DIRAC
- Analysis Productions: A declarative approach to ntupling
- LbMCSubmit: A new flexible and scalable request submission system for LHCb simulation
- Integrating LHCb workflows on Supercomputers: State of Practice
- ARC and the EuroScienceGateway project
- Migration to WebDAV in Belle II Experiment
- The Cherenkov Telescope Array Observatory workflow management system
SUPPORTED RESOURCES
Computing Resources

Where to run the jobs

DIRAC embeds plugins to interact with various computing resources:

- **Clusters (Plugin: SSHCE + BatchSystem interface):** Orchestrated by a BS. Generally accessed through an SSH/GSI SSH tunnel.

- **Grid Sites (Plugin: HTCondorCE, ARCCE/AREXCE):** Clusters with specific policies, accessed through a CE.

- **HPC Sites:** Clusters with additional constraints. More details in this presentation: Integrating LHCb workflows on Supercomputers: State of Practice

- **Cloud resources (Plugin: CloudCE):** More details on this poster: Standardizing DIRAC’s Cloud Interfaces.

- **Volunteering resources (Plugin: BOINCCE):** BOINC Volunteer resources.

- **Locally (Plugin: LocalCE).**
Storage Resources

Where to store the results

DIRAC also support various protocols to interact with storage resources:

- **S3**: e.g. AWS and CEPH.
- **SRM, XROOT, HTTPS, GSIFTP**: using GFAL2.
- **RFIO protocol (deprecated)**
- **"File"**: abstraction of the local storage as an SE.
- **DIP**: DIRAC Custom Protocol.

Several abstractions of the same physical endpoint are possible (Multi-protocol). Storage occupancy information can be fetched from BDII or WLCG Accounting.
SYSTEMS
Basics of DIRAC WMS

- **Push model**: Error-prone, but reintroduced to exploit HPCs with no external connectivity.
- **Pull model**: Pilot-Job paradigm is the most used way of submitting jobs.
- **Vacuum model**: (HLT Farm, VAC).
Data Management System (DMS): Transferring data to storage interfaces

Basics of DIRAC DMS

- LFNs (Logical File Name): unique identifier of a file within DIRAC.
- LFNs may have physical replicas, stored in SEs.
- LFNs are registered in catalog(s). There exist multiple implementations of catalogs. Several of them can live in parallel:
  - DIRAC File Catalog: full replica and metadata catalog.
  - Plugins for DIRAC TS, LHCb Bookkeeping, RUCIO.
- DMS integrates FTS3 to schedule and monitor efficient transfer of large amounts of data between SEs.
Some VOs using DIRAC would like to use Rucio as DMS (and maybe some VOs using Rucio would like to use DIRAC WMS).

- Discussions started at the 8th DIRAC workshop (May 2018).
- Few developments done on both sides: integration of (multi-VO) DIRAC with (multi-VO) Rucio. **In Progress**
- Since January 2021 Belle2 uses DIRAC and Rucio (from LCG file catalog to Rucio FC).

If interested, we will held a DIRAC & Rucio Workshop, details here: [https://indico.cern.ch/e/DR23](https://indico.cern.ch/e/DR23)
Transformation System: Job productions and datasets management

Used to automate common tasks related to production activities

- Production: "Data Processing" transformation (e.g. Simulation, Merge, DataReconstruction...). It ends up creating jobs in the WMS.

- Data Manipulation: transformation to replicate, or remove data from storage elements. It ends up creating requests in the RMS (Request Management System), which feeds the DMS.

The Transformation System is finely tuned and can manage millions of jobs and files daily.
Accounting & Monitoring

Accounting

- For historic data: jobs, pilots, data operations, storage
- MySQL backend, visualized in DIRAC WebApp

Monitoring

- Real Time monitoring and not only
- OpenSearch backend, visualized in Kibana, Grafana and (partially) DIRAC WebApp

Settling with Grafana as next-gen visualization tool, also for historic data
From X509 certificates to OIDC Tokens

DIRAC v8 rationalizes many aspects to AuthN, AuthZ, Tokens and OAuth2 support.

- Support new Identity Providers (IAM & EGI CheckIn) New: v8.0.
- Use tokens (and/or proxies) to submit jobs/pilot-jobs New: v8.0.
- Use tokens (and/or proxies) to interact with storage resources (v8.x)
### Highlights of the latest developments

- Transitioning service communications from DIPS (DIRAC in-house protocol) to HTTPS. **In Progress**

- Replacing the in-house task queue system (Executors) by Celery, a widely used task queue system. **In Progress**

- Centralizing logs coming from Pilot-Jobs. More details on this poster: [Improved Pilot Logging in DIRAC](#) **In Progress**
DEVELOPMENT
## Development, testing and deployment

<table>
<thead>
<tr>
<th>Section</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Development</strong></td>
<td>~5 FTE as core developers, a dozen contributing developers.</td>
</tr>
</tbody>
</table>
| **Testing** | Tests, certification, integration process is a daily work.  
  • We use GitHub Actions, and Jenkins for some bits.  
  • We run certification hackathons every two weeks. |
| **Deployment** | Puppet profiles used for long time, helm chart available on request. |
CONCLUSION
Additional resources

• Documentation: dirac.readthedocs.io.
• Code documentation: here
• Dev and DevOps issues: on GitHub
• Ops and general questions: GitHub discussions
• Bi-weekly developers meetings (and/or hackathons): BILD

DIRAC & Rucio Workshop

• 16-20 October 2023 in KEK, Japan.
• Registration and details: https://indico.cern.ch/e/DR23

Questions? Comments?
DIRAC extensions

“Horizontal” extensibility
- For specific requirements

“Vertical” extensibility
- Community driven

Each project is independently versioned

“pip install VOWebAppDIRAC”
Presentation

• Stores info on the status of Resources (e.g. SEs)
• An autonomic computing tool evaluates a few policies to determine the status of the resources. E.g.:
  • Space left < threshold → ban for writing
  • Endpoint in downtime in GocDB → ban r/w
• DIRAC SEs states are sync-ed from DIRAC RSS to Rucio via a DIRAC agent
RSS

A generic system, which can be used for queueing (also) DMS operations. Operation types: ReplicateAndRegister (e.g. using FTS), RemoveFile/RemoveReplica, ...

and the Request Executing Agent will... execute the requests.