



# Monitoring DIRAC Components

Pujan Mehta

**Mentors:** Christophe Haen, Zoltan Mathe, Federico Stagni

# About DIRAC



**DIRAC** (Distributed Infrastructure with Remote Agent Control) Interware is a complex, open source software which is used to perform large-scale tasks such as:

- Workload Management
- Data Management
- Orchestration of Distributed Resources

..... and much more.



**DIRAC** is majorly used in the **LHCb** experiment and **LHCb** is also **DIRAC's** initiator and main contributor. Apart from this **DIRAC** is also used by several Virtual Organizations.



# GSoC Project Tasks

- Write Technical documentation on the working of **DIRAC's gMonitor** object.
- Migrate **DIRAC's Service** and **Agent** components to support **ElasticSearch** backend.
- Create Visualizations based on this on the **WebAppDIRAC**.
- Migrate **DIRAC's RequestManagementSystem** to support **ElasticSearch** backend.
- Write **Integration** and **Performance** tests.
- Develop code compatible in **Python 3**.



# Migrating DIRAC's service and agent components to Elasticsearch

- First in this task I migrated DIRAC's service component which covered monitoring fields such as **Running Threads**, **Max File Descriptors**, **Active Queries**, **Number of Connections**, etc.
- Then I reported the **Service Response Time** to the **ElasticSearch** backend.
- Working on agent component I covered monitoring fields like an agent's **cycle duration** and **number of cycles** performed by the agent because they are one of the active **DIRAC** components.
- Then I worked on adding new visualizations to the **WebApp** and running an existing performance tests whose results can be found [here](#).



# Migrating DIRAC's RMS to ElasticSearch

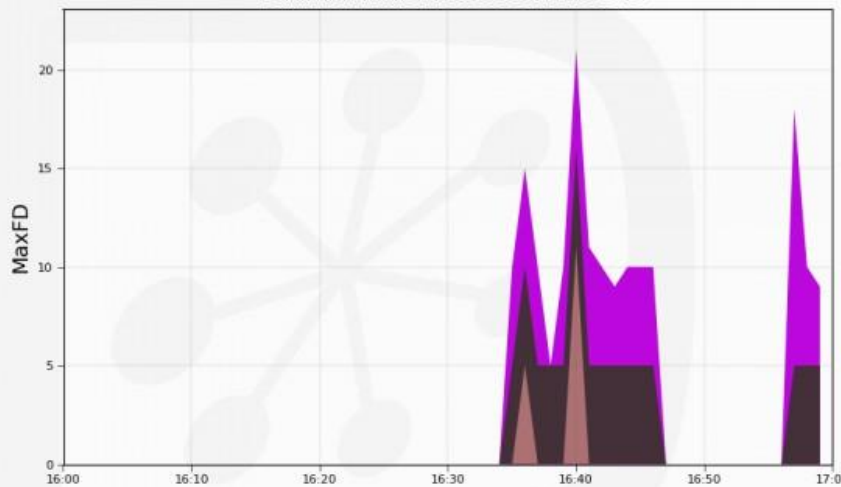
- In this task, first I worked on designing an **ElasticSearch** type such that it can encompass all the use cases of the **gMonitor** object inside **DIRAC's RequestManagementSystem** and **DataManagementSystem/Agent/RequestOperations/**.
- After this I worked on migrating the code to support **ElasticSearch**.
- Then worked on creating visualizations on **DIRAC's WebApp**

This system is used to perform **Data Management** tasks such as **File Transfers**, **Replications**, **Removals**, etc. on various storage resources.

Our main motive, behind this was to monitor the behaviour of the execution pattern of these tasks.

# Demo of the Visualizations

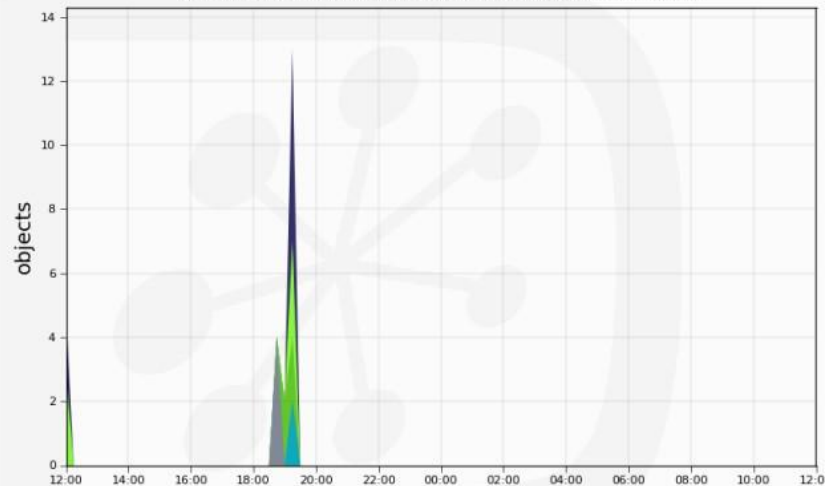
Max File Descriptors by componentName  
60 Minutes from 16:00:00 to 17:00:00 UTC



Configuration/Server	45.8%	Framework/SystemAdministrator	0.0%
Framework/ComponentMonitoring	44.6%	Framework/Monitoring	0.0%
Monitoring/Monitoring	9.5%	Framework/UserProfileManager	0.0%

Generated on 2019-07-04 17:00:00 UTC

File Operations  
24 Hours from 2019-08-21 12:00 to 2019-08-22 12:00 UTC



ReplicateAndRegister	33.3%	PutAndRegister	16.7%	RemoveReplica	8.3%
RemoveFile	25.0%	MoveReplica	16.7%		

Generated on 2019-08-22 12:04:02 UTC

# Technologies Used

- Python
- ElasticSearch
- Docker
- Kibana



elasticsearch



docker



kibana



**Thank You**