The LHCb DIRAC-based production and data management operations systems

F Stagni, P Charpentier

PH Department, CH-1211 Geneva 23 Switzerland

E-mail: federico.stagni@cern.ch, philippe.charpentier@cern.ch

On behalf of LHCb Collaboration

Abstract. The LHCb computing model was designed in order to support the LHCb physics program, taking into account LHCb specificities (event sizes, processing times etc...). Within this model several key activities are defined, the most important of which are real data processing (reconstruction, stripping and streaming, group and user analysis), Monte-Carlo simulation and data replication. In this contribution we detail how these activities are managed by the LHCbDIRAC Data Transformation System. The LHCbDIRAC Data Transformation System leverages the workload and data management capabilities provided by DIRAC, a generic community grid solution, to support data-driven workflows (or DAGs). The ability to combine workload and data tasks within a single DAG allows to create highly sophisticated workflows with the individual steps linked by the availability of data. This approach also provides the advantage of a single point at which all activities can be monitored and controlled. While several interfaces are currently supported (including python API and CLI), we will present the ability to create LHCb workflows through a secure web interface, control their state in addition to creating and submitting jobs. To highlight the versatility of the system we present in more detail experience with real data of the 2010 and 2011 LHC run.

1. Introduction

The Production system transforms physics requests into physics events, implementing the LHCb computing model. The heart of the system is the Transformation System. The LHCbDIRAC Transformation system is a key component for all the distributed computing activities of LHCb. It is a reliable, performant, and easy to extend system. Together with the Bookkeping system and the Requests Management system, it provides what is called the Production System. This paper explains how the Production System is developed and used, giving an architectural and an operational perspective. It will also show experience from 2010 and 2011 Grid processing.

This paper is organized as follows: section 2 introduces DIRAC and LHCbDIRAC. Section 3 explains the architecture of the production system, as a whole, introducing the concepts used, together with a short description of their technical implementation. Within section 4, we give a brief summary of how the operations team uses the system for its daily activities. Section 5 shows some results from past year's run, and final remarks are given in section 6.

2. LHCbDIRAC

DIRAC (Distributed Infrastructure with Remote Agent Control) [1] is a community Grid solution. Developed in python, it offers powerful job submission functionalities, and a developer-friendly way to create services and agents. A DIRAC service exposes an XML-RPC implementation; a DIRAC agent is a stateless light-weight component (comparable to a cron-job).

DIRAC has been initially developed as a LHCb-specific project, but many efforts have been made to re-engineering it into a generic framework, capable to serve the distributed computing needs of a number of Virtual Organizations. After this complex reorganization, the LHCb-specific code resides in the LHCbDIRAC extension while a core, VO-agnostic, DIRAC project has been disentangled. In this way, other VOs, like Belle II [2], or ILC/LCD [3] have developed their custom extensions to this core DIRAC framework. DIRAC is a collection of sub-systems, each constituted of services. Sub-systems are, for example, the WMS (Workload Management System) or the DMS (Data Management System). Each system comprises a generic part, and a VO-specific part.

LHCbDIRAC extends DIRAC to handle all the distributed computing activities of LHCb. Such activities include real data processing (reconstruction, stripping and streaming), Monte-Carlo simulation and data replication. Other activities are groups and user analysis, data management, resources management and monitoring, data provenance, accounting for user and production jobs. While DIRAC and LHCbDIRAC follow independent release cycles, every LHCbDIRAC is built on top of an existing DIRAC release.

3. The Production System

The LHCb distributed computing activities include production and non-production activities. Within this paper, we have no interest in non-production activities, which include *user analysis*, and *monitoring and testing*. The production activities can be split between data manipulation, and data management. Data manipulation comprises *Reconstruction*, *Reprocessing*, *Stripping* (which is the name LHCb gave to a selection of events), *Calibration* and *Montecarlo simulations*. Data management activities are usually a direct consequence of the production activities of Data Manipulation. They include: *Replication*, *Archival* and *Removal*. These data management activities share a large fraction of the DIRAC and LHCbDIRAC code used for data manipulations.

Within this section, we introduce the concepts used in the system, and their implementations: DIRAC workflows, the DIRAC and LHCbDIRAC transformation system, the LHCb bookkeeping and production requests system.

3.1. DIRAC Workflows

A workflow is, by definition, a sequence of connected steps. DIRAC provides an implementation of the workflow concepts in one of its "Core" packages, with the declared scope of running "complex" jobs, i.e. jobs who run one application after another, whose input/outputs are usually directly connected to each other. The implementation comes in the interchangeable formats of an XML file, or an extended python dictionary. All production jobs are described using a DIRAC workflow.

As can be seen in figure 1, each worfklow is composed of steps, and each step include a number of modules. These workflow modules are connected to python modules, that are executed in the specified order by the jobs. Parameters can be specified at any level: workflow, step, and even modules.

Figure 1. Concepts of a worfklow: each workflow is composed by steps, that include modules

3.2. DIRAC Transformation System

The DIRAC Transformation System is used for handling "repetitive" work. It has two main uses: the first is for creating job productions, and the second for data management operations. When a new "Production Jobs" transformation is requested, a number of transformation tasks are dynamically created, based on the input files (if present), and the "plugin" specified. A "plugin" specifies the way the tasks are created, and can decide where the jobs will run, or how many input files can go into the same task. Each of the tasks pertaining to the same transformation will run the DIRAC workflow specified when the transformation is created.

A task, when inside the transformation system, is not yet a Grid job nor a data management operation. For this last step, agents are defined, that upon inspections of the Transformation System tables, will submit the tasks either to the Workload Management System, or the Data Management System. The Transformation System, unless when being used just for simulation activities, can not live as a stand-alone system: whenever there are input data to be handled, external Metadata and Replica catalogs are needed. For LHCb, the Metadata and Replica Catalogs are the LHCb Bookkeeping [4], presented in the next section, and the LFC catalog [5]. Figure 2 shows some components of the system implementation.

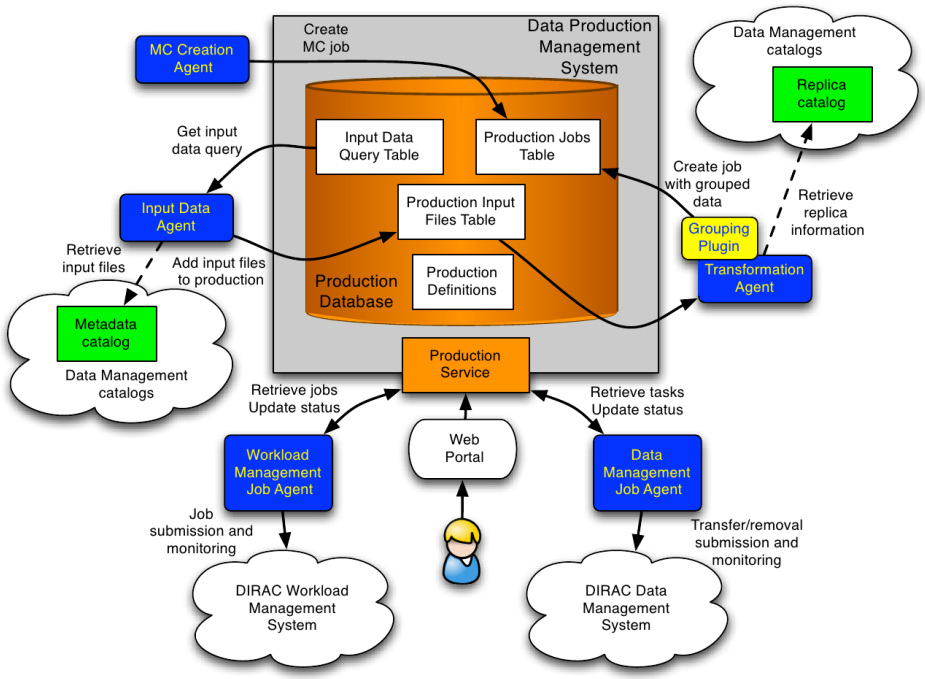

Figure 2. Schematic view of the DIRAC Transformation System

3.3. LHCb Bookkeping

The LHCb Bookkeping System is integrated within LHCbDIRAC, and it is the LHCb data recording and data provenance tool. The Bookkeeping System is widely used inside the Production System, by a large fraction of the members of the Operations team, and by all the physicists doing data analysis within the collaboration. Being recognized as a key system by the collaboration, its design and implementation were subject to important modifications throughout the years.

The bookkeeping is not necessarily a tool for doing distributed computing. Among its functions, users and machines are retrieving datasets for analysis and productions, together with metdata like data taking and simulation conditions, event types and file types. The bookkeeping is a read-only database for general users.

Unlike the other DIRAC system, where MySQL is the prime choice as RDBMS, the CERN Oracle backend has been chosen instead. Figure 3 shows the main components within this system.

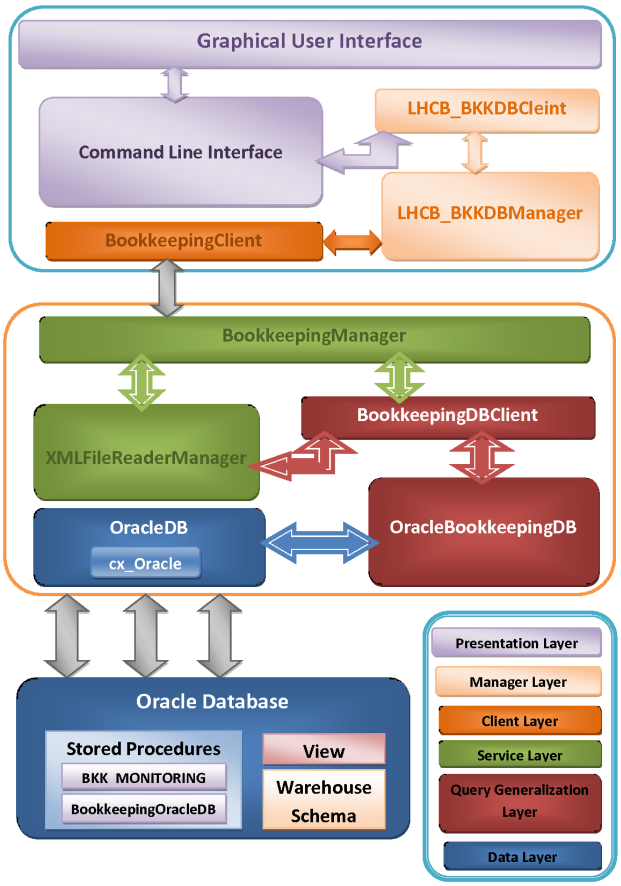

Figure 3. Schematic view of the LHCbDIRAC Bookkeping System. The complexity of the system deserves an explanation per se. A number of layers are present, as shown in the legend in the bottom-right corner of the figure. Within this paper, we just observe that the data recorded represent the first source of information for production activities within LHCb.

3.4. LHCbDIRAC Production Request System
The *Production Request System* [6] is a way to expose the users to the production system. It is an LHCbDIRAC developement, and represents first of all a formalization of the processing activities. A production is a combination of *steps*, and a production request can be created by any users, provided that there are formal *steps* definition in the steps database. Figure 4 shows this simple concept.

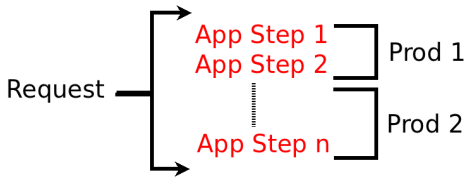

Figure 4. Concepts of a request: each request is composed by a number of steps

Creating a step, a production request, and subsequently launching such production is done using a web interface, integrated in the LHCbDIRAC web portal, in a user-friendly way. The production request web page also offers a simple monitoring of the status of each request: for example, the percentage of processed events, as requested, is reported and publicly available. Such information is also used, for simulation productions, to trigger their automatic completion.

4. Operating Productions
Putting all the pieces together, operationally means a workflow composed of the following steps:

(i) *Application managers* create usable steps, where all the options are set;
(ii) *Production Managers* combine the available steps to create production requests;
(iii) The same *Production Managers* create real productions starting from such requests, using production templates;
(iv) The Productions are followed by the computing shifters, the *GEOC (Grid Expert On Call)* and by the same Production Managers.

A Data Manager creates bulk replication, deletion or archival of datasets using the transformation system. There are tools provided for massive final checks, that for example allows to know if the correct number of replicas are registered in the catalogs and on the storage services. Many interfaces are provided: GUI/CLI used in all steps. In figure 5 all the concepts discussed are put together.

5. The system at work in 2010 and 2011
In figure 6, we summarize the results from 2010 processing, using four plots. A few notable points that we can extrapolate are:

- *Many user jobs*: analysis jobs from users represent a large fraction of the distributed activities of 2010.
- *MC productions*: the use of the production system was mostly used for simulation campaigns.
- *Activities*: even if peaks of 50k jobs/day were reached, the system sustained an average of 20k jobs/day.
- *Use of Tier 0*: CERN resources were enough for one fifth of the total amount of resources used.

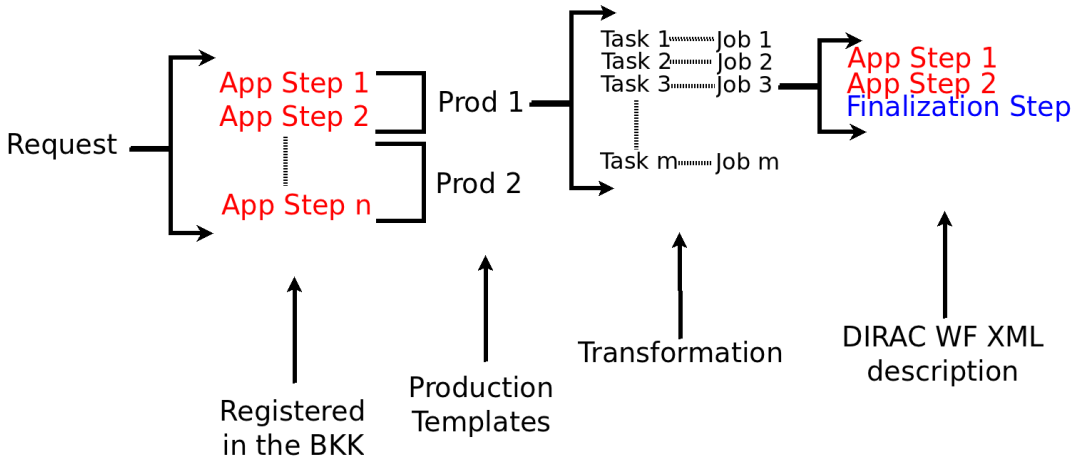

Figure 5. Putting all the concepts together: from a Request to a Production job

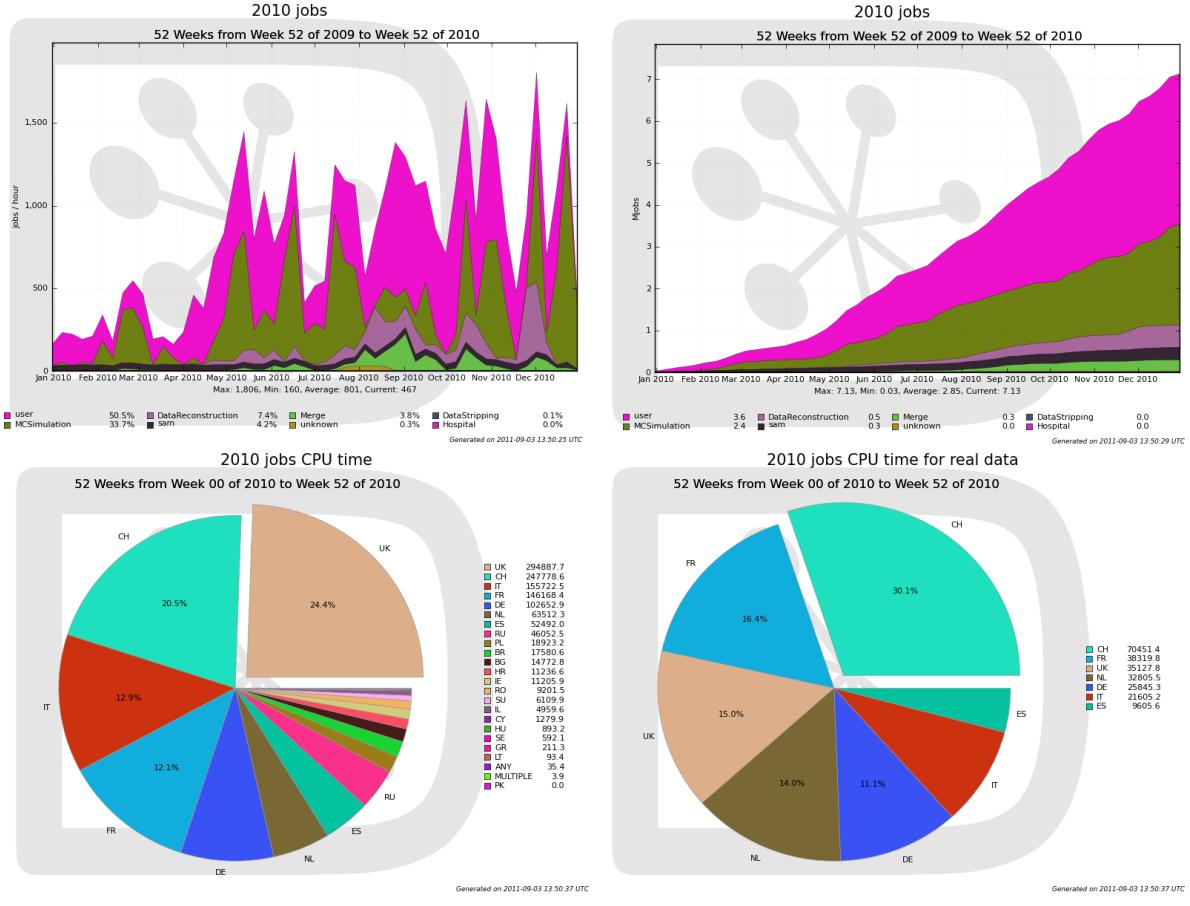

Figure 6. Some plots showing the results of data processing in 2010. Clockwise: the rate of jobs per hour in the system by job type, the cumulative number of jobs managed throughout the year, the percentage of CPU used for real data processing, and the percentage of CPU used by country by the totality of the jobs.

Figure 7 shows four plots related to part of the 2011 activities. What can be seen is that:

- *MC productions*: a huge simulation campaign has started at the beginning of the year, and continued steadily through the year.
- *Activities*: the system sustained an average of 40k jobs/day, which is twice what was done during the previous year.
- *Efficiency*: the wasted CPU time has been less than 5%. This is due to the high efficiency of the LHCb software, and LHCbDIRAC itself.
- *Use of resource*: mostly due to the simulation campaign, all Tier 2 have been intensively used.

It has to be noted that these plots for 2011 covers only the first 34 weeks of the year. During the last months of 2011, a *reprocessing* activity started. Such activity used resources from Tier 1 and Tier 2.

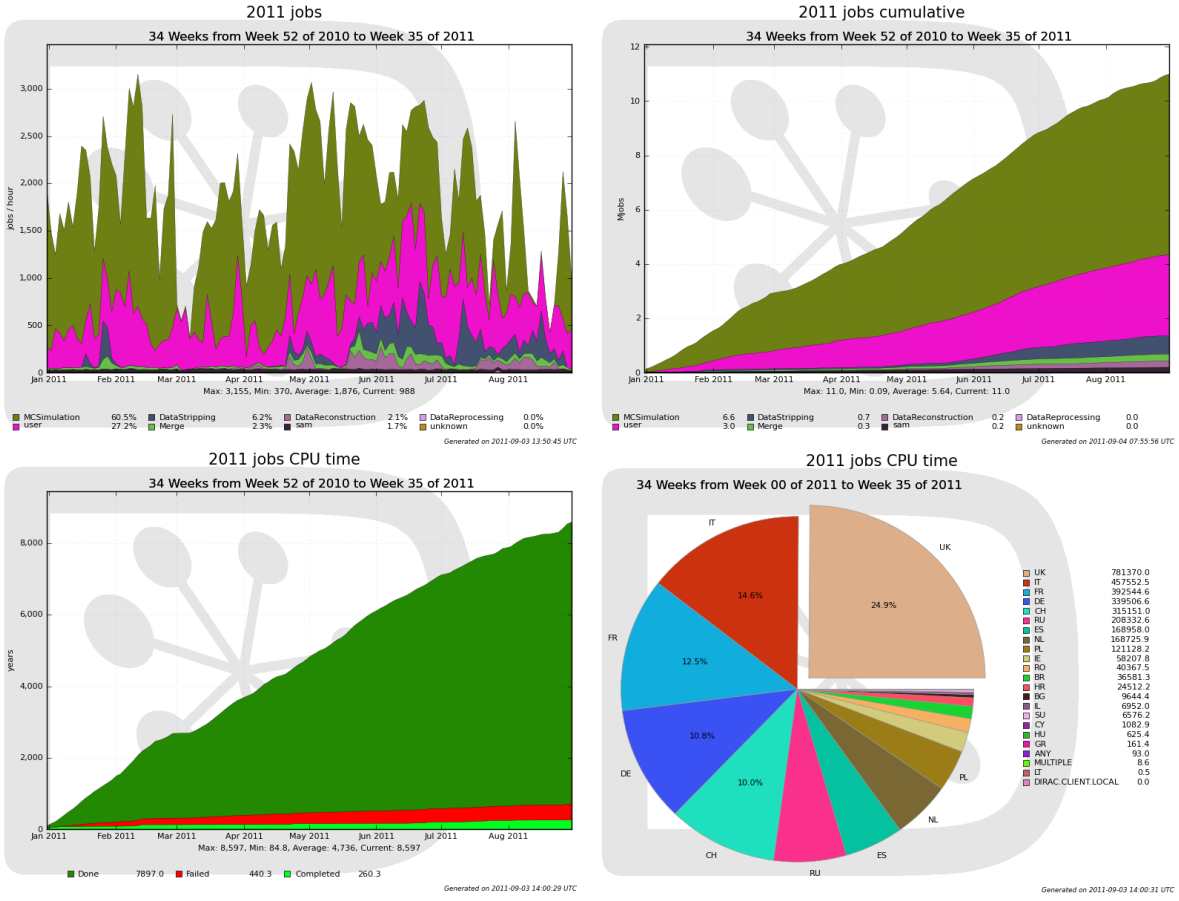

Figure 7. Some plots showing the results of data processing in 2011. Clockwise: the rate of jobs per hour in the system by job type, the cumulative number of jobs managed throughout the year, the percentage of CPU used by country by the totality of the jobs, and the cumulative number of jobs by their final major status.

6. Conclusions

DIRAC has proved to be a scalable, and extensible software. The production system is vital for a great part of LHCb distributed activities. The system first started to be used with simulation campaigns, and it is now used for Data Reconstruction, Reprocessing, Stripping, Calibration

activities. It is also used for datasets manipulations like data replication or data removal. During the last years, we have seen how the number of jobs handled by the Production system steadily grew, and represents now more than half of the total number of jobs created. User analysis jobs still represent a good fraction of the jobs handled by the system, and since they have a priority over production jobs, when a large number of them are queued, they may represent an operational burden, slowing down if not completely blocking part of the production activities. For this reason, pressure has recently put on analysis working groups to integrate as much as possible their activities in the production framework. This would allow the operations team to better control the stress put on computing and storage resources, depending on the relative priorities of the various scheduled production activities. Working Groups will also gain advantages when coming to data provenance, recording and distribution of the produced data.

References

[1] Tsaregorodtsev A, Bargiotti M, Brook N, Ramo A C, Castellani G, Charpentier P, Cioffi C, Closier J, Diaz R G, Kuznetsov G, Li Y Y, Nandakumar R, Paterson S, Santinelli R, Smith A C, Miguelez M S and Jimenez S G 2008 *Journal of Physics: Conference Series* **119** 062048 URL `http://stacks.iop.org/1742-6596/119/i=6/a=062048`

[2] Abe T and Al 2010 Belle II Technical Design Report URL `http://xxx.lanl.gov/abs/1011.0352v1`

[3] Barish B and Al Global Design Effort URL `http://www.linearcollider.org/GDE`

[4] Lanciotti E and Mathe Z 2009 Lhcb: The lhcb data bookkeeping system

[5] Bonifazi F, Carbone A, Perez E D, D'Apice A, dell'Agnello L, Dllmann D, Girone M, Re G L, Martelli B, Peco G, Ricci P P, Sapunenko V, Vagnoni V and Vitlacil D 2008 *J. Phys.: Conf. Ser.* **119** 042005

[6] Tsaregorodtsev A and Zhelezov A 2009 Lhcb: Managing large data productions in lhcb