# **Offline Processing in the Online Computer Farm**

L G Cardoso<sup>1</sup>, C Gaspar<sup>1</sup>, O Callot<sup>2</sup>, J Closier<sup>1</sup>, N Neufeld<sup>1</sup>, M Frank<sup>1</sup>, B Jost<sup>1</sup>, P Charpentier<sup>1</sup>, G Liu<sup>1</sup>

<sup>1</sup>CERN, European Organization for Nuclear Research,1211 Genève 23, Switzerland <sup>2</sup>LAL, Laboratoire de l'Accelerateur Lineaire, 91898 Orsay, France

E-mail: luis.granado@cern.ch, clara.gaspar@cern.ch, olivier.callot@cern.ch, joel.closier@cern.ch, niko.neufeld@cern.ch, markus.frank@cern.ch, beat.jost@cern.ch, philippe.charpentier@cern.ch, guoming.liu@cern.ch

**Abstract**. LHCb is one of the 4 experiments at the LHC accelerator at CERN. LHCb has approximately 1500 PCs for processing the High Level Trigger (HLT) during physics data acquisition. During periods when data acquisition is not required or the resources needed for data acquisition are reduced most of these PCs are idle or very little used. In these periods it is possible to profit from the unused processing capacity to reprocess earlier datasets with the newest applications (code and calibration constants), thus reducing the CPU capacity needed on the Grid. The offline computing environment is based on LHCbDIRAC (Distributed Infrastructure with Remote Agent Control) to process physics data on the Grid. In DIRAC, agents are started on Worker Nodes, pull available jobs from the DIRAC central WMS (Workload Management System) and process them on the available resources. A Control System was developed which is able to launch, control and monitor the agents for the offline resources (e.g. DBs) and in case of change of the accelerator planning it can easily return the used resources for online purposes. This control system is based on the existing Online System Control infrastructure, the PVSS SCADA and the FSM toolkit.

#### 1. Introduction

LHCb is one of the 4 experiments at the LHC accelerator at CERN. The LHC delivers 40 MHz to LHCb [1], which then needs to be reduced to a manageable rate to be stored and analyzed. The reduction of the data rate is achieved by the means of a two levels trigger system. A first level trigger, hardware based, reduces the rate of accepted events to 1 MHz. These events then go through a second level trigger called High Level Trigger (HLT) which finally reduces the rate to 5 KHz. HLT is completely software based and runs on a dedicated computer farm with 1470 PCs totaling over 16.000 cores.

Outside data taking periods, the CPU needs for High Level Triggering are very low and most of the available CPU power sits idle. To increase the efficiency of the resource usage and process data faster, during these periods, the HLT farm nodes can be used as worker nodes for DIRAC (Distributed Infrastructure with Remote Agent Control) and process physics data. To effectively manage the offline data processing on the online farm the network had to be setup in order to be able to retrieve data from the DIRAC Workload Management System (WMS) and upload the process data back to the DIRAC servers, and a control system was developed to properly manage the DIRAC agents on the worker



nodes as well as to manage the available resources according to online/offline processing requirements. This system will be detailed in the following sections.

Figure 1. LHCb Data Acquisition System with data transfer rates

Average rate to tape 5 kHz

### 2. Online infrastructure

#### 2.1. HLT Farm setup

The online farm for HLT is composed by 1470 PCs, distributed by 56 subfarms. These subfarms are logically divided in the Control System by the rack letter where they are installed (e.g. HLTA, HLTB) and then by rack number (e.g. HLTA01, HLTA02). Each subfarm (divided by rack letter) is composed of 10 subfarms (divided by rack number) with 27 PCs each, with a mix of older and newer PCs. Each of the subfarms is controlled by a controller PC – with PVSS installed – which manages the HLT tasks on the HLT nodes. These controller nodes are also connected to a top level HLT control node, which manages the availability and allocation of the subfarms for the global Experiment Control System (ECS).

# 2.2. Network setup

The LHCb Online network is a private network and accessible only from the CERN network via special gateway (GW) hosts. LHCb has a dedicated 10-Gigabit link into the CERN LCG backbone network, which is used for data transfer to CERN CASTOR for permanent storage. The data transfer is done by a few nodes which have the direct access to the CERN LCG network.

#### 2.3. Offline computing environment

The DIRAC System [2] is a specialized system for data production, reconstruction and analysis of the data produced by HEP experiments (e.g. LHCb). It follows the Service Oriented Architecture (SOA) and its components can be grouped in the following 4 categories: Resources, Services, Agents and Interfaces.

Of particular interest for our application are the Agents. Agents are light and easy to deploy software components which run as independent processes to fulfill one or several system functions. Agents run in different environments and they watch for changes in the service states and react accordingly by initiating actions like job submission or result retrieval. Agents can run as part of a job executed on a Worker Node as so called "Pilot Agents".

International Conference on Computing in High Energy and Nuclear Physics 2012 (CHEP2012) IOP Publishing Journal of Physics: Conference Series **396** (2012) 032052 doi:10.1088/1742-6596/396/3/032052

The Workload Management System (WMS) is a central DIRAC component to support the production of data for the LHCb Experiment. It includes a central Task Queue and a network of light agents.

The light Pilot Agents are deployed close to the computer resources which are presented in a uniform way to the DIRAC WMS. A check of the sanity of the operational environment in which the jobs will be executed is then done, before the pull of the real workload from the central Task Queue.

# **3. DIRAC on the online infrastructure**

#### 3.1. Infrastructure

The ONLINE DIRAC Control system is composed by a Linux PC, where the PVSS Scada and the control system software are implemented which connects to each of the PCs of the HLT subfarms. This PC also acts as the main DIM (Distributed Information Management) node for the required communication connections to the worker nodes. It is also connected to the main HLT control PC, in order to be able to check and set the availability of the HLT subfarms for online or offline usage.

# 3.2. Network setup

As LHCb has a private network, all the nodes in HLT farm have only private IP addresses, they are not accessible from outside LHCb. In order to allow the HLT nodes access the data stored in CERN castor, masquerade NAT (Network Address Translation) is deployed in the store nodes which have the access to the CERN LCG network. NAT is designed for IP address conservation, which allows private IP networks use a public IP addresses to communicate with hosts outside. With masquerade NAT, the HLT nodes can copy the data from CASTOR and do the data processing.

# 4. Implementation

The ONLINE DIRAC Control System was developed in PVSS and uses the same tools (DIM, FMC, FSM, etc.) as the rest of the Control System to guarantee its integration and a coherent look and feel.

# 4.1. DIM

DIM (Distributed Information Management) [4] is a tool developed at CERN designed to provide a common communication layer to the processes involved with the different tasks of the Online System. It is based on the client/server paradigm. Three instances are needed to establish communication with DIM:

- A server, which serves the data in the form of services and commands to be subscribed;
- A client, which subscribes to the data from the services required and sends commands to be executed remotely;
- A Name Server which acts as an intermediary at the startup of the communications at startup the servers register their available services and commands with the Name server and the client gets the available services information from the Name Server as well. After the initial connection is established, the communication between the client and the server is direct.

Both the FMC and FSM toolkit implement DIM communication.

# 4.2. Farm Monitoring and Control (FMC)

A set of tools to monitor and manage several parameters on the PCs was developed for the LHCb experiment. This set of tools is composed of several servers which run on each node and are able to be accessed by remote systems, communicating via DIM (Distributed Information Management System). These servers can monitor and manage several parameters of the PCs where they run such as resources usage, process management or operating system monitoring. Of particular interest to the Online DIRAC Control System application is the tmSrv (Task Manager Server) [3]: this server is able to

International Conference on Computing in High Energy and Nuclear Physics 2012 (CHEP2012) IOP Publishing Journal of Physics: Conference Series **396** (2012) 032052 doi:10.1088/1742-6596/396/3/032052

launch and monitor processes on a remote node, attributing to each process a unique identifier (UTGID) which can be tracked via a subscribed DIM service on another system.

# 4.3. FMC Server and DIM subscriptions

The Online DIRAC Control System relies on the Task Manager FMC server to start the DIRAC script on the worker nodes. For this reason, this server needs to running on all the worker nodes which are to be used to process the offline data, which point to the DIM DNS node used for the DIM subscriptions on the Control System node. Normally these servers could be started as services in the worker nodes for ease of usage. However, to avoid unneeded usage of resources, these servers can be started via a PVSS panel, only on the nodes which are to be used for offline data processing. For each worker node, the DIM service which provides a list of the currently started processes via the Task Manager Server is subscribed into a PVSS datapoint (per node); this allows the monitoring at any time of the UTGID (User assigned unique Thread Group Identifier) and the PID of the DIRAC agent tasks running on each node and the monitoring for task lifetime and maximum number of running DIRAC agents. Also for each node, a DIM command can be sent to the Task Manager server in order to start/stop/kill a currently running task on this node.

#### 4.4. DIRAC Script

Basically, the task which is run on each worker node is an Agent which is querying the DIRAC Workload Management System to check if there is some task to be executed. If it gets a job, the execution will take place in the local disk where the input data, if any, will be downloaded and the output will be written. At the end of the task, the ouput(s) will be uploaded to the Storage located in the Computer Center. During the execution of the task, information is sent back to the DIRAC monitoring to follow the progress of the job.

#### 4.5. PVSS Ctrl Manager

In a usual Control Application the nodes could receive a command and perform it in parallel. However, in this case this cannot happen as if it were so, the DIRAC WMS database would be overwhelmed with requests at startup.

A PVSS Control Manager was developed to handle the launching, monitoring and management the DIRAC agents on the worker nodes. This manager is always running and is responsible for:

- connecting and disconnecting the monitoring of the worker nodes to make sure that no unneeded connections are consuming resources;
- managing the delay between starting of DIRAC agents to make sure there are no starting agents within a specified time frame, so the DB doesn't get overwhelmed with requests simultaneously;
- balancing the load between the allocated farms by maintaining an ordered list of nodes where the agents are to be started and sorting this list according to pre-defined load balancing rules;
- monitoring the connection of the needed DIM services.

The availability of jobs to process from the DIRAC Work Management System is monitored by verifying the running time of the agents on the worker nodes. The control manager is able to increase or decrease the delay between the start of two agents by monitoring the average running time of the last 10 started agents. In case there are no jobs to be picked up from the DIRAC WMS, the agents will end rather quickly and the time until the start of the next agent is increased. Similarly if the agents run for a long time, the delay will be decreased and agents will be started more often (provided that there are available "slots" to start agents). This ensures a reduced number of requests to the DIRAC database in the case there are no jobs to process.

# 4.6. FSM

The Control System is interfaced by a Finite State Machine implemented using the FSM toolkit [5] developed at CERN, interfaced with PVSS.

The FSM toolkit provides two types of objects that can be used to build a FSM tree:

- Device Units model the real devices in the control tree (whether they are physical components or software components). Each device is modeled as a FSM and have a state and a set of actions available for each state. They are the lowest nodes on a FSM tree.
- Control Units group Device Units (and/or other Control Units) into logical useful segments. They also have a state and set of actions available. The actions are propagated down the tree to its children and its state depends on the state of its children.

The FSM tree of the Online DIRAC Control System models the available subfarms in the same way that they are setup for the HLT system (divided by Rack Letter and then by Rack number). On each subfarm there are the Device Units corresponding to the several PCs that exist in that subfarm, as well as a Device Unit that models the allocation state of that particular subfarm. The granularity of the system should be considered the subfarm (by Rack Number) as a whole subfarm needs to be allocated in order to start the processing. The worker nodes in which the processing will occur can be selected individually; however, the whole subfarm to which that node belongs to will be allocated for offline processing and thus unavailable for online purposes.

Four types of objects were defined for the FSM:

- Top (Control Unit) Groups all the subfarms divided by Rack Letter and all the subfarms belonging to a rack, divided by Rack Number
- Subfarm (Control Unit) Groups all the subfarms belonging to a rack, divided by Rack Number
- Subfarm Allocator (Device Unit) Controls the allocation of a subfarm for offline purposes
- Worker Node Info (Device Unit) Controls the running of the DIRAC agents on each worker node

4.6.1. FSM Objects. The existing States for the subfarms Control Units are:

- UNAVAILABLE The farm is allocated for other usage and not available for Offline data processing
- NOT\_ALLOCATED The farm is free
- ALLOCATED The farm is currently allocated for Offline data processing
- RUNNING The farm has currently at least one node processing Offline data
- STOPPING The farm has currently at least one node processing Offline data, however all the nodes are set not to start new agents when the ones that are currently running terminate
- ERROR

These objects allow a logical grouping of the resources as well as the desired granularity for operation.

For the Worker Node Info Device Unit the available states are:

- UNAVAILABLE The node is allocated for other usage and not available for Offline data processing
- IDLE The node is free
- RUNNING The node has currently offline data processing agents running
- STOPPING The node has currently offline data processing agents running, however no other agent will be started on this node when the ones that are currently running terminate



**Figure 2.** Finite State Machine of the most relevant objects. (a) Subfarm Control Unit FSM; (b) Worker Node Device Unit FSM;

When a worker node is available and sent the action GOTO\_RUNNING, a series of checks is done on the worker node to guarantee that the required environment is properly set on it. First, it is checked whether the required FMC Task Manager Server is running so that the processes can be properly started and monitored. Then it is checked which type of CPU is available on this node, so that the maximum number of slots to start agents can be set. Finally it is checked whether the working directories on this node are properly created and cleaned. In case any of the check fails, a warning is produced on the FSM User Interface.

### 4.7. User Interface

The User Interface was developed in PVSS, to maintain a coherent Look and Feel with the other Control Software of the Experiment. It presents synoptic panels where the user can easily see the current status of the system and it is interfaced via the usual FSM interface developed for PVSS. The user can easily see how many farms are available for Offline Processing, allocate/deallocate and start/stop running the processes in the worker nodes. Statistics of the current usage of the system are available and it can be easily seen how many subfarms and nodes are allocated as well as the number of running agents per node and subfarm. In each node it is also possible to see for how long the current agents have been running.



**Figure 3.** Online DIRAC Control system User Interface and DIRAC Monitoring web interface

International Conference on Computing in High Energy and Nuclear Physics 2012 (CHEP2012) IOP Publishing Journal of Physics: Conference Series **396** (2012) 032052 doi:10.1088/1742-6596/396/3/032052

#### 5. Conclusion

The implemented Control System is a tool that facilitates the usage of the available resources of the experiment for the processing of Offline Data. The development based on PVSS and the FSM toolkit helps the adoption and usage by the people of the experiment by providing a familiar look and feel as the rest of the Control System

The efficiency of the online farm usage is improved by having an easy tool to automatize the required setup of the system, and the launch and monitoring of the offline data processing; At the same time, GRID computing resources are reduced.

The system is now in production in the LHCb control room.

#### References

- [1] Frank M, Gaspar C, Van Herwijnen E, Jost B, Neufeld N, Cherukuwada S and Stoica R 2008 *J. Phys. Conf. Ser.* **119** 022023
- [2] Tsaregorodtsev A et al. 2008 J. Phys. Conf. Ser. 119 062048
- [3] Bonifazi F, Carbone A, Galli D, Gaspar C, Gregori D, Marconi U, Peco G, Vagnoni V M and Van Herwijnen, E.2008 *IEEE Trans.Nuc. Sc.* **55** 370-378
- [4] Gaspar C and Franek B 2006 IEEE Trans.Nuc. Sc. 53 3
- [5] Gaspar C, Donszelmann M and Charpentier Ph 2000 Computer Physics Communications 140 102-109