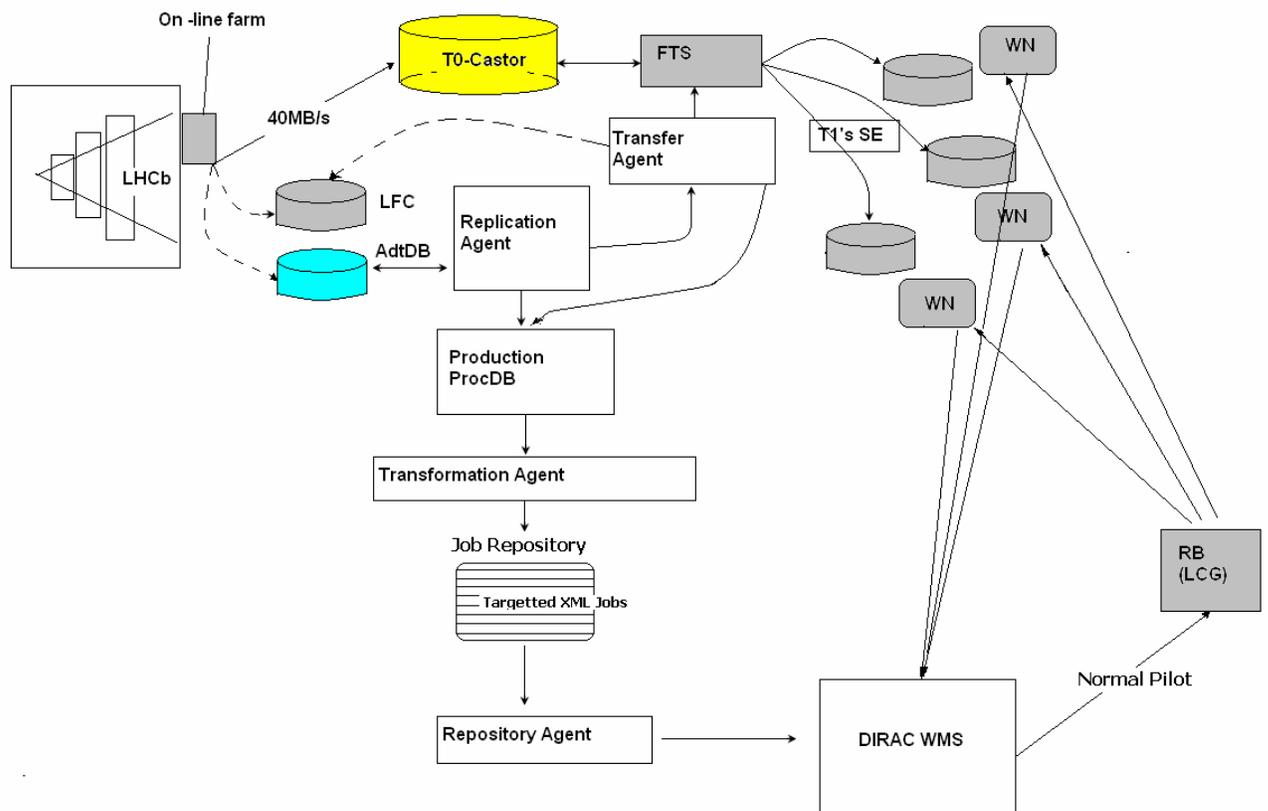


DC06 infrastructure

Introduction

With this informal note we would like to present the current mechanism used on DC06 activity. The aim of the document is not intended to give to the reader a full overview of LHCb computing framework, or an extensive description of the computing model or a detailed description of Dirac (for which we remind to other write-ups and publications) but it is rather an eye-of-bird view of various components and their interactions.



The figure schematizes the system currently used by LHCb for DC06; this same mechanism, with few other improvements, is intended to be the final draft of the system that will be used for data reprocessing, as declared in the Computing Model Document. The aim of DC06 is also to validate this mechanism.

The flow

1. Data is preliminary processed from the online farm and then stored on the T0 Castor at CERN at a rate of about 40MB/s. These data are registered both on the LCG File Catalog (accessed afterward by final jobs from the WNs) and also on

the special LHCb Data Base: **AdtDB**. The AdtDB stands for **AutoDataTransfer DB** and it is basically a DataBase (with a MySQL backend) that stores:

- The LFN/site_name pair: for instance /lhcb/production/DC04/v2-lumi-6/DST/00000793_00005697_5.dst [CERN-tape]
- And an object called **transformation**

The transformation is in turn a complex object that contains a name- a mask to be applied to data and a Regular Expression that are applied for this particular transformation on the name of the lfn,

2. Every 10 seconds a **Replication Agent** runs and polls the AdtDB looking for new files to be processed. It checks if the files in the database fit with specific transformation and - on top of the information in its own configuration files - once the number of files that fit the transformation reaches a certain limit (configurable) it replicates those files to the various T1s accordingly to a share (also configurable). For sake of simplicity we brought the following example.

Dirac.ini

→*replicationagent.ini*

GROUPSIZE=300

SOURCESITE=CERN-tape

TARGETSITE=CERN-tape, RAL-tape, PIC-tape, CNAF-tape,

IN2P3-disk, SARA-tape, FZK-tape

CERN-tape=10

SARA-tape=0

CNAF-tape=0

PIC-tape=10

FZK-tape=10

RAL-tape=20

IN2P3-disk=10

The example above shows the interesting part of the configuration file that it is typically used by any Dirac agent. Once the Replication Agent gets a number of files fitting a given transformation equal to the GROUPSIZE value, this special Agent builds a special Dirac XML jobs, whose type is *transfer*. By using the above example and translating words into numbers once there are 300 files that fit with a special transformation, these files are replicated to various TARGETSITES with a ratio proportional to the one assigned to each in the conf file. Again in the example PIC will receive 50 files while RAL 100 and so on. The XML jobs is submitted to the Dirac WMS (as any other job is) and is picked up by another special agent called **TransferAgent** (running on *lxgate34*).

3. The Transfer Agent, once pulled the job (in a similar way the **Job Agent** does from whatever WN on LCG) breaks it in several bulk requests to the various sites accordingly the specified share set by the Replication Agent and submits them to FTS. This agent runs on behalf of production manager credentials; there are features that enable to easily switch identity. The activity of the transformation agent is described more in detail in other document/articles available on web.
4. Once the transfer requests are successfully executed by FTS and the replica is correctly registered onto the LFC, the Transfer Agent inserts a new entry in the **Processing Data Base** which has the same format as any other entry in the

AdtDB: i.e. [lfn][site-name]. CERN is treated differently because data is already there and not transfers are then needed.

5. Another agent (***Transformation Agent***) runs and polls the Processing DB parsing for each site entries that fits with another "*transformation*". This last allows creating automatically Dirac XML jobs and is defined by the Production Managers. It takes into account the number of input files, the version of various applications to run and so forth. Once a given threshold (configurable through a dirac.ini configuration file) is reached, the Transformation Agent creates a DIRAC XML job to be submitted to that particular site and saves it into a dedicate area called ***Job Repositor***. This area is also filled up also by other mechanisms (like the ***Dirac Console***).
6. A ***Repository Agent***, as soon as a job appears in the Job Repository submits it to the WMS, updating the statuses and collecting the output of it. The targeted job is then now available on the ***Task Queue*** for being picked up by generic Pilot submitted by the Dirac WMS. With respect of normal production jobs they present a higher priority that translates into an escalation of the central Task Queue.