CRAB: the CMS distributed analysis tool development and design

D. Spiga\textsuperscript{a}, S. Lacaprara \textsuperscript{b}, W. Bacchi\textsuperscript{c}, M. Cinquilli\textsuperscript{a}, G. Codispoti\textsuperscript{c}, M. Corvo\textsuperscript{d}, A. Dorigo\textsuperscript{e}, A. Fanfani\textsuperscript{c}, F. Fanzago\textsuperscript{d}, F. Farina\textsuperscript{f}, O. Gutsche\textsuperscript{g}, C. Kavka\textsuperscript{h}, M. Merlo\textsuperscript{f}, L. Servoli\textsuperscript{a}.

\textsuperscript{a}University and INFN Perugia, \textsuperscript{b}INFN Legnaro, \textsuperscript{c}University and INFN Bologna, \textsuperscript{d}CERN, \textsuperscript{e}INFN Padova, \textsuperscript{f}INFN Milano-Bicocca, \textsuperscript{g}FNAL, \textsuperscript{h}INFN Trieste.

Starting from 2007 the CMS experiment will produce several Pbytes of data each year, to be distributed over many computing centers located in many different countries. The CMS computing model defines how the data are to be distributed such that CMS physicists can access them in an efficient manner in order to perform their physics analysis. CRAB (CMS Remote Analysis Builder) is a specific tool, designed and developed by the CMS collaboration, that facilitates access to the distributed data in a very transparent way. The tool's main feature is the possibility of distributing and parallelizing the local CMS batch data analysis processes over different Grid environments without any specific knowledge of the underlying computational infrastructures. More specifically CRAB allows the transparent usage of WLCG, gLite and OSG middleware. CRAB interacts with both the local user environment, with CMS Data Management services and with the Grid middleware.

1. Introduction

CMS is one of the four experiments that will collect data at LHC. The CMS detector has 15 millions of channels; through them data will be taken at a rate of some TB/s, of which just some will be selected to be written on disk. To catch those data there is an on-line selection system (trigger) that will reduce the frequency of data taken from 40 MHz (LHC frequency) to 100 Hz (writing data frequency), that means 100 MB/s and 1 PB data per year. The use of the grid instruments chosen by LCG and OSG projects allows to solve a complicated problem: the access to the data and the distributed resources by CMS users. The Computing Model of CMS defines an hierarchic architecture of the grid sites with one Tier-0 site (CERN) that is linked with the CMS Data Acquisition System, some Tier-1 sites and many Tier-2 centers. There are also Tier-3 sites mainly department resources. The data will pass first through the Tier-0 site and will be processed and transferred to the lower level sites. For every transfer there is a manipulation and selection of the data in order to reduce the dimension and to get just the interesting part of those.

2. Distributed analysis with CRAB

The data analysis on a distributed environment is a complex computing task because the used data have a dimension of hundred Megabytes, that makes the data transfer not convenient for just some small analysis. Also the other heterogeneous instruments are not local, but distributed all around the world. CRAB is the official CMS analysis software that easily interfaces the user with the grid environment hiding the system complexities. Following the analysis model it allows an easy access to the data distributed over the grid in a very transparent way and the user is not required to have any deep knowledge about the grid. Infact CRAB simplify the process of CMS analysis allowing to process officially published data and hiding as much as possible the grid complexity to the final user, so that remote data can be accessed with the same facility of local data: CMS user sends through this tool his analysis code to the site where the selected data are.

CRAB needs to be installed on the User Interface (UI) -which is the user access point to the grid-, it supports any CMSSW (the CMS software framework) based executable, with any modules/libraries, including the user provided ones, and finally it deals with the output produced by
the executable. From a user point of view, the basic steps of the workflow of this software are:

- **Job Creation**: interaction with data discovery services (DBS and DLS) and with the user environment; the task is split into smaller jobs and the input sandbox is prepared.
- **Job Submission**: interaction with Resource Broker, Workload Management System and proxy services to submit jobs to sites matching the user requirements.
- **Job Status**: check the status of the jobs using BOSS mechanism.
- **Job Output**: retrieval of the job output from the grid (output sandbox); also the output can be transferred to a nearby Storage Element or to another one specified by the user.

Other functionalities are: the possibility to get information about aborted jobs; to kill jobs and to resubmit failed jobs.

3. CRAB improvement

The CRAB evolution aims at:

- Automatizing as much as possible the interaction with the grid (submission, resubmission, output retrieval, etc.)
- Reducing the unnecessary human load, moving all possible actions to server side, reducing to a minimum those on client side.
- Improving scalability of the whole system.

To reach the above objectives it has been developed and it is still under development a server with which CRAB can communicate. Then CRAB will be used, also that to submit jobs directly to the grid, to submit jobs through the server. This client-server implementation is transparent from the user point of view. What is most important is that the server automatizes operations like job/task tracking, resubmission, output retrieval and error handling. Its architecture adopts a modular software approach with independent components that are implemented as agents communicating through an asynchronous and persistent message service. Parts of the server are also a GridFTP server and a Proxy Delegation service that allows to store a valid proxy for the user inside the server.

The current server components that can already makes a complete distributed analysis are:

- **DropBoxGuardian**: checks drop-box for new user tasks and new proxy arrivals.
- **ProxyTarAssociator**: associates task to right user and localizes the task configuration.
- **CrabWorker**: submits jobs to the grid using the submitter module of CRAB (EDG, gLite, gLite-bulk, condor-g support).
- **TaskTracking**: keeps all the general information about tasks under execution.
- **Notification**: notifies the user by an e-mail when his task is ended and the output has been already retrieved.
- **JobTracking**: tracks the status of every job.
- **ErrorHandler**: performs a basic error handling.
- **JobSubmitter**: resubmits single jobs when needed.
- **RSSFeeder**: provides channels to forward information about the server.

Some other components that will be soon ready are: disk management, kill component, job/task status via web.

REFERENCES

1. CRAB project http://cmsdoc.cern.ch/cms/ccs/wm/www/Crab
2. CRAB twiki https://twiki.cern.ch/twiki/bin/view/CMS/CrabServer
3. LCG Project http://lcg.web.cern.ch/LCG
4. The CMS experiment http://cmsdoc.cern.ch
5. BOSS project http://boss.bo.infn.it