

# CRAB usage and jobs flow monitoring

D. Spiga<sup>a\*</sup>, L. Servoli<sup>a</sup>, L. Faina<sup>a</sup>

<sup>a</sup>INFN and University of Perugia

## Abstract

CMS is one of the four experiments expected to take data at LHC. Order of some PetaBytes of data per year will be stored in several computing sites all over the world.

The collaboration has to provide tools for accessing and processing the data in a distributed environment, using the grid infrastructure. CRAB (CMS Remote Analysis Builder) is a user-friendly tool developed by INFN within the CMS, to help the user to submit a job to the grid, featuring the input data discovery, the job creation, the job submission, the monitoring of the job status, the output retrieval and, finally, the handling of the user output.

In this abstract we describe how we monitor the use of CRAB, by means of an applications which collects the jobs informations from submission time to output retrieval time, and send them to a DataBase. We also present some analysis of the data collected during the second half of 2005, aimed at understanding the performance of the whole system and at finding eventual bottlenecks, in order to improve the system.

## INTRODUCTION

The Compact Muon Solenoid (CMS) [1] is one of the four particle physics experiments that will collect data at the Large Hadron Collider (LHC) [2] now under construction at CERN in Switzerland.

With the expected rate of about 150Hz on disk, the large amounts of events (few PByte per year) that will be stored and processed when the detector starts collecting data, needs a distributed computing and data access solution.

Collected data will be distributed in many regional centers participating to the CMS collaboration in order to made them available and analysable from a large number of physicist in many geographically distributed sites.

The most promising solution seems to be the grid paradigm, so the CMS collaboration choose a distributed architecture based on some of its components. A combination of generic grid tools, provided by the European LCG (LHC Computing Grid) [3] and the American OSG (Open Science Grid) [4] projects, and specialized CMS tools will be required to manage the data and optimize the use of the distributed computing resources.

In this paper it's shown the usage and the job flow of CRAB, a tool developed by INFN (Istituto Nazionale Fisica Nucleare) within the CMS to allow a user to analyse the events stored in a remote center, using a prototype

of a job flow monitoring system.

In the first section a brief description of CRAB [5] and its main functionalities is given; in the following section a description of the monitoring system idea and of the technical implementation with an overview of the quantities monitored is provided; in the last section some results obtained analysing the data stored during the monitoring activity are presented.

## CRAB (CMS REMOTE ANALYSIS BUILDER)

CRAB is a CMS specific tool written in Python and developed within INFN in order to open the grid to the masses. Installing it on the User Interfaces, CMS users can have an easier access to all data produced and available, using grid middleware.

In figure 1 the CRAB job flow with the various interactions between grid and CMS services is shown. CRAB will shield the user from the grid complexities; he just need to develop his analysis code in an interactive environment using the program for CMS Reconstruction (ORCA), and to choose a dataset to be analysed.

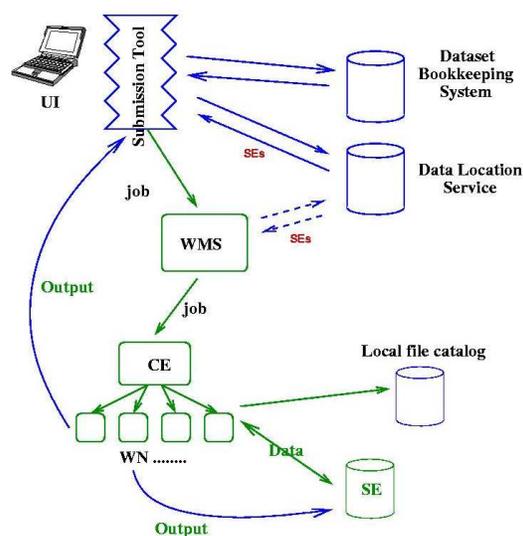


Figure 1: Interaction between the grid and CMS services for a typical CRAB job flow.

CRAB main functionalities are:

- Input data discovery, to determine the Computing Element (CE) of sites storing data. They are found querying CMS specific services such as central (RefDB) and local (PubDBs) databases.

\* Work supported by Consorzio CRESCI

- Packaging of user code obtained creating a `tgz` archive containing the user code and the `lib` and `data` structures.
- Job creation, consisting in the definition of a Wrapper (`sh` script) to perform the sanity check on remote resources (WN), to define the CMS framework to run an ORCA executable and, finally, to handle the job output.
- Job Description Language (JDL) file creation, where all the parameters to be passed to the Resource Broker, (RB) (e.g CEs name) to drive resources matchmaking process, are defined.
- Job splitting according to user requests
- Job submission to the grid, via Workload Management System command (`edg-job-submit`).
- Monitoring of job status, check the status of jobs on user demand.
- Output retrieval and handling to offer the possibility to choose to copy to the UI or to a generic Storage Element (SE) or to any host with a `gsiftp` server (e.g. CASTOR).

## CRAB MONITORING TOOL

The development and deployment of CRAB increases the need to monitor and to analyse the use of the tool. The first monitoring system has been designed in order to achieve two main goals; to show in real time some informations such as:

- RATE of CRAB-Jobs Submission;
- Dataset and Computing Element Usage;
- Provenance of Jobs;

and then to answer, with an off-line analysis, to questions like:

- How efficient is the service;
- How many users are using CRAB (and how);
- Which patterns of data access are emerging (which data are used and where);
- Which are the failures/problems of the service;
- How to improve the user support;

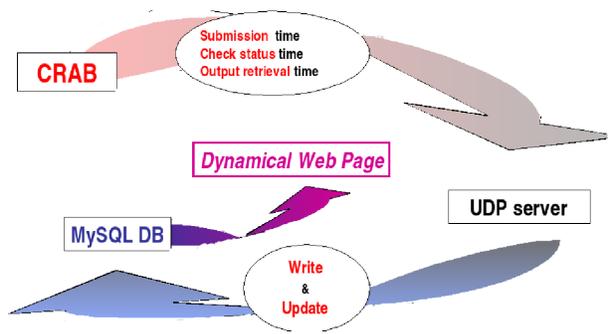


Figure 2: Technical implementation of the monitoring tool.

### Technical Implementation

The Technical implementation of the monitoring tool is based on UDP tecnolgy (fig. 2). At three diffent points of its workflow (submission, check status, output retrieval), CRAB sends UDP packets, containing the informations, to an UDP server which process the data and fills a MySQL database. In order to satisfy the request for a real-time monitoring, a php web page which shows the results of the query to the database storing data is created. The web page is then automatically updated to display always the latest informations.

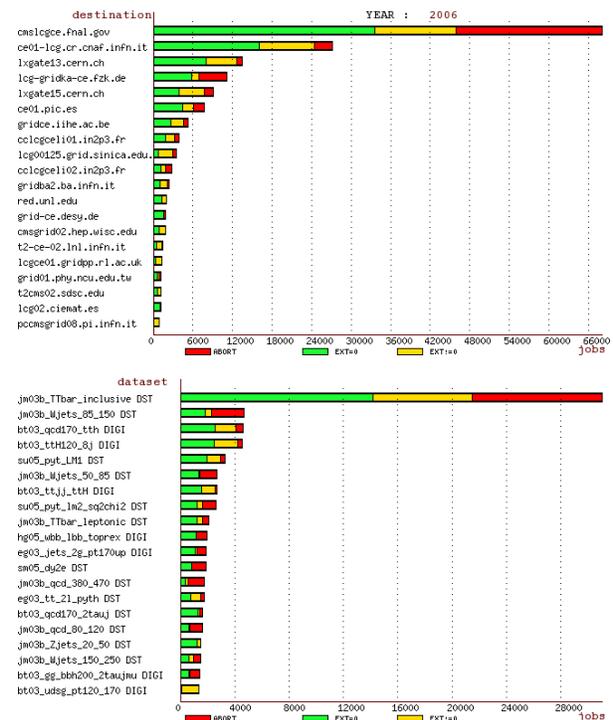


Figure 3: Distribution of most frequently used Computing Elements where CRAB jobs run (top) and of most frequently requested Dataset/owner by CRAB jobs (bottom). Color codes are: ORCA Exit Code equal to zero (green); ORCA Exit Code different from zero (yellow); jobs aborted for grid problems (red).

## USE OF COLLECTED DATA

As mentioned the two main goals of the monitoring are to be able to show some information on real time querying database and to analyse stored data.

### *On line informations Display*

The first information available online is the daily, weekly, monthly and total rate of jobs submitted using CRAB, but it's also possible to know where the jobs run, which input data they use and from which User Interfaces have been submitted.

The histograms related to the destination of jobs and their input data are organized to give informations about the quality of the results. As shown in fig. 3 (respectively CE and dataset/owner usage), each bar represents the total number of jobs and it is divided into three categories: jobs that produce ORCA Exit Code equal to zero (green), those that produce ORCA Exit Status different from 0 (yellow), and jobs that could not run due to grid problems (red) They are useful to have fast feedback on the behaviour of the system.

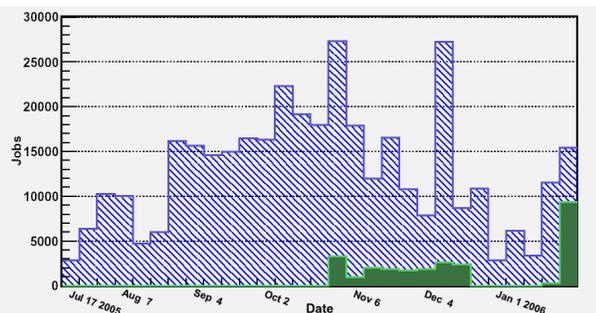


Figure 4: Weekly Rate of CRAB jobs. Dashed blue represent LCG jobs, green color OSG jobs.

### *Off line analysis*

The same data collected by the monitoring tool are also available off line, giving the possibility to analyse them to understand in depth the performances of the system. The study of the time evolution of several quantities allows to

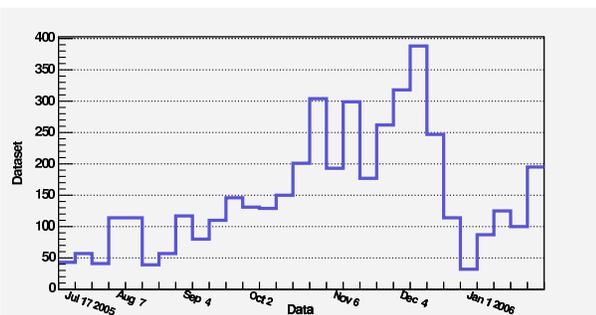


Figure 5: Weekly request rate for different datasets/owner

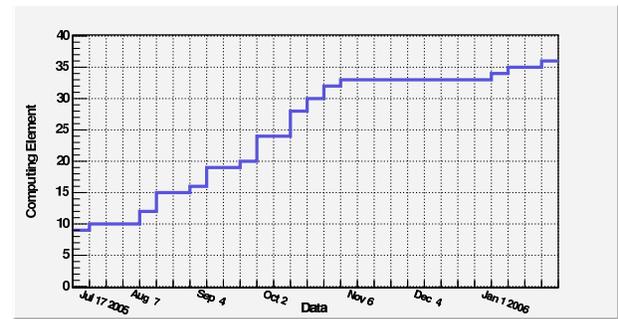


Figure 6: Time integral of the different computing element used

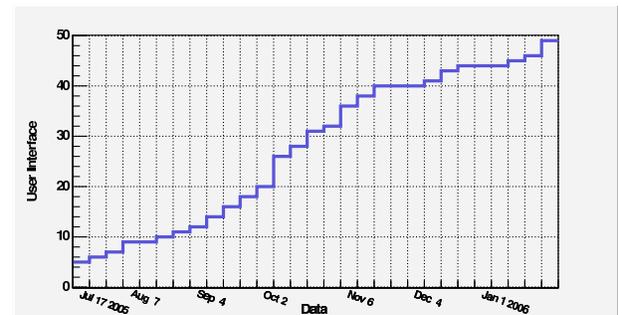


Figure 7: Time integral of the different user interfaces used

draw some conclusions on the use and the efficiency of the tools.

The data show that from July 2005 to January 2006 about 400000 CRAB-jobs were submitted on the grid; in figure 4 the weekly submission rate is plotted, differentiating among the LCG (dashed blue) and the OSG (green) grid infrastructure used. It can be noticed that the rate is substantially constant among the period examined (2000 jobs per week) excepts of the startup period (July 2005) and of the end of the year.

The plot showing the weekly request rate for different datasets (fig 5) gives an indication of how many datasets (currently about 390) have been requested by the users to make their analysis.

To know how many sites offer their resources it's possible (fig. 6) to plot the time integral of the different Computing Element where CRAB jobs have run (currently about 40)

The same kind of histogram (fig. 7) reporting the number of different User Interfaces that have used CRAB to submit jobs on the grid, allows to understand the diffusion among users of the tool (currently about 50).

From the results shown in the previous histograms it's possible to see the increasing number both of sites that are storing and making available data for CMS distributed analysis and of users that are using CRAB to analyse simulated data to obtain physics results.

Finally looking at the weekly success rate for CRAB jobs (fig. 8), the ratio between the jobs which don't fail for infrastructure reasons and the total number of jobs, it's possible

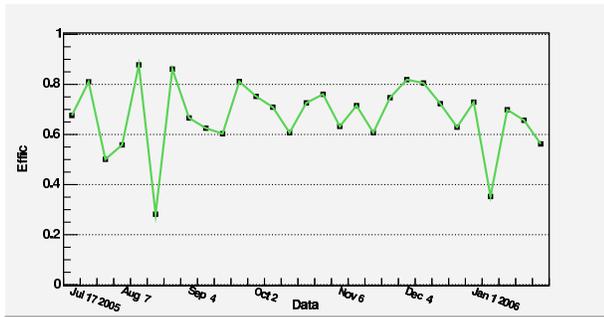


Figure 8: Weekly success rate for CRAB jobs.

to measure the middleware efficiency, that is currently of the order of about 70%.

## CONCLUSION

CRAB monitoring tool was developed and deployed starting data recording from July 2005. It was implemented a web page in order to show in real time informations related to the use of CRAB, which data are used, destinations and provenance of jobs.

The collected data offer also the possibility to perform some off line analysis whose results allow to understand problematics and features related to CRAB. The first results reached by this monitoring system shows that CMS use of the grid paradigm to analyse data is increasing steadily. The data management could also use the results in order to understand which datasets need to be duplicated and where.

Lastly the average failure rate due to the middleware is monitored continuously; the conclusion is that the whole system is working in a reasonable way.

## ACKNOWLEDGEMENTS

We wish to thank Hassen Rihai who contributed to the development and deployment of the web page php scripts and the CRAB team who provided support during the implementation of the monitoring.

## REFERENCES

- [1] The CMS experiment: <http://cmsdoc.cern.ch>
- [2] The Large Hadron Collider Conceptual Design Report CERN/AC/95-05
- [3] LCG Project: <http://lcg.web.cern.ch/LCG> and LCG Technical Design Report, CERN-TDR-01 CERN-LHCC-2005-024, June 2005
- [4] OSG Project: <http://www.opensciencegrid.org>
- [5] CRAB project: <http://cmsdoc.cern.ch/cms/ccs/wm/www/Crab>
- [6] ORCA project: <http://cmsdoc.cern.ch/orca>
- [7] PubDB project: <http://cmsdoc.cern.ch/swdev/viewcvsv/viewcvsv.cgi/OCTOPUS/PubDB>

- [8] Job Description Language HowTo December 17, 2001 available at <http://server11.infn.it/workload-grid/docs/DataGrid-01-TEN-0102-02-Document.pdf>