

Table of Contents

5.8 Job Monitoring with CMS Dashboard.....	1
Contents.....	1
CMS Dashboard provides a web interface for Job Monitoring.....	1
How to follow the progress of your tasks.....	1
If you see any discrepancy between information in the Dashboard and the output of the Crab status command.....	3
Using Dashboard Interactive Interface.....	3
Review Status.....	4

5.8 Job Monitoring with CMS Dashboard

Complete: 
Detailed Review Status

Contents

- CMS Dashboard provides a web interface for Job Monitoring
- How to follow the progress of your tasks
- If you see any discrepancy between information in the Dashboard and the output of the Crab status command
- Using Dashboard Interactive Interface

CMS Dashboard provides a web interface for Job Monitoring

Most of the CMS Job Submission Systems, including CRAB and PA, are instrumented to send monitoring information to the CMS Dashboard. In addition to reports of the CMS Job Submission Systems, Dashboard collects information from the Grid Monitoring systems. Monitoring data is stored in the central database and there is a web interface running on top of it and allowing CMS users to follow the progress of their jobs. For the Dashboard monitoring you do not need to setup any environment. The only thing you need is a web browser. In case you see a problem, please submit a bug in savannah:
<https://savannah.cern.ch/projects/dashboard> [↗](#) mail to [dashboard-support@cernNOSPAMPLEASE.ch](mailto:dashboard-support@cern.ch)

How to follow the progress of your tasks

- After submission of your task to the CRAB server you get back the Dashboard task monitoring link, which you can follow in order to get information about progress of your task.
- If you want to find information about multiple tasks submitted during a particular time range you can enter the application via the entry task monitoring page:

<http://dashb-cms-job-task.cern.ch/dashboard/request.py/taskmonitoring> [↗](#)

- Choose your identity in the "Select a User" window and the time window to define the tasks submitted during a given time range. You should get at the screen the list of all your tasks submitted over the time range you have chosen.

As a rule, on the Dashboard UI, user identity is defined by user name and family name. But it is not always the case. User identity is retrieved from the Grid certificate subject and depends on its format. To check your Dashboard user identifier, go to the Dashboard interactive job monitoring page:

<http://dashb-cms-job.cern.ch/dashboard/request.py/jobsummary> [↗](#)

Click on the bar with the 'analysis' label and sort by user. Looking on the user names listed next to the bars or in the table below, you will find out your Dashboard identifier. At the task monitoring page you see the list of all your tasks with the distribution of jobs by their current status.

You can bookmark the link to task monitoring application containing information only about your tasks:

<http://dashb-cms-job-task.cern.ch/dashboard/request.py/taskmonitoring#action=tasksTable&usergridname=USERNAME>

where USERNAME is your Dashboard identifier

You can also bookmark the link to a particular task:

<http://dashb-cms-job-task.cern.ch/dashboard/request.py/taskmonitoring#action=taskJobs&usergridname=USERNAME&taskmonid=TASKMONITORID>

where TASKMONITORID is project name in CRAB.

- The front page shows the overview of all tasks submitted during the selected time range (3 days by default) in table and graphical form.

Every page is reloaded every few minutes providing the most up-to-date information.

- Clicking on the information icon next to the task name you get the **meta information** of a particular task, like name of the input dataset, CMSSW version to be used, time stamp when task had been registered in the Dashboard.
- Clicking on the number of jobs corresponding to a given status gives you a detailed information of all jobs of a chosen category: Grid job id, identifier of the job inside the task, how many times the job had been resubmitted, site where job is processed, time stamps of the job processing (UTC).
- In case when the job was **resubmitted multiple times**, clicking at the number at the "Submission Attempts" column allows you to see all resubmissions corresponding to a given job inside the task. Here we are referring not to the resubmissions triggered by resource broker, but to resubmissions done by the user.
- The application provides various graphs showing distribution of jobs by site, time, failure reason, and graphs showing time consumed by the task, the graph showing progress of the task in terms of processed events. The plots either generated by default on an appropriate page or can be selected clicking on the "Plot Selection" link in the table. Graphs can be zoomed-in and zoomed-out.
- Jobs listed in the "**Successful**" category are those which had accomplished properly from the application point of view and for which Dashboard did not get evidence that the job had been aborted by the Grid.
- The jobs listed as "**Failed**" are those which either failed from the application point of view or had been aborted by the Grid or cancelled by the user/crab_server. By failure from the application point of view we mean non 0 status from the CMS application or problems related to the saving of the output files at the storage element, or problems while sourcing of the CMS environment at the site. To discover the **reason of failure** of the particular job, click on the number of jobs in the "Failed" category, you get the list of all failed jobs with the Grid status and application exit code. Moving cursor at the status value you get more detailed reason of failure.
- Please, pay attention, for back navigation between the task monitoring pages do not use "back" button in the browser, use buttons provided on the task monitoring pages.
- On the task page showing the overview of all jobs belonging to a task next to the task name you see the small Dashboard logo icon. Clicking on it you get the **bar plot** of distribution of jobs of the task by a chosen attribute, like site, computing element or resource broker. Sometimes this distribution can help you to find the problem of a given site, CE or resource broker and can allow you to exclude the problematic service or site when you resubmit your jobs. See more details in the section "Using

Dashboard Interactive Interface".

If you see any discrepancy between information in the Dashboard and the output of the Crab status command

Sometime you can notice the discrepancy of the information in the Dashboard and the Crab status command.

- Dashboard does not have user credentials and can not directly query Grid Logging and Bookkeeping system to get status about a particular job. It relies on the job status reports sent to the dashboard either from the jobs themselves or job status information of the Grid monitoring systems like RGMA or ICRTM. That is why if you see non-consistent data in Crab and Dashboard related to the Grid status of a given job, you should believe Crab.
- On the other hand Dashboard gets real time information about jobs which is reported to the Dashboard from the jobs running at the worker nodes. So when you see that according to the Dashboard the job had terminated while Crab still considers the job to be running, it means that the job had already finished on the worker node and sent it's exit status, while Grid Logging and Bookkeeping system did not yet update the job status. If you see that a delay in update of crab status for the job which had terminated according to Dashboard takes too long (more than half a hour), the problem can be related to the Grid services.
- For some sites, e.g. T3_US_FNALLPC, nothing is reported to dashboard until the job is done.

You can follow the CRAB3 Troubleshooting guide for more on how to troubleshoot your job and contact crab support.

Using Dashboard Interactive Interface

One of the purposes of the Dashboard Interactive Interface is to show the correlations of the job failures or inefficiencies in processing (pending too long in the queue for example) with a particular site or Grid service like Resource Broker.

When you see that all jobs of a particular task are failing and it is not clear to you whether it is a problem of your code or the problem related to the site misconfiguration, Dashboard Interactive Interface can help you to find it out.

- First thing you can check is whether jobs of other users are failing at the same site with the same failure code. Go to:

<http://dashb-cms-job.cern.ch/dashboard/request.py/jobsummary>

By default on the interactive user interface you see all jobs submitted during the selected time window. If you tick the checkbox 'terminated' then you would see all jobs which are currently either in pending or running status or those which had been terminated from the date selected as the beginning of the time range to now regardless the time when the jobs had been submitted. Be aware that all dates in Dashboard including UI are UTC. You can sort the jobs by user, site, computing element, resource broker, application, task. Clicking at any bar of the plot would allow you to sort the subset of jobs shown in a particular bar by various attributes. Clicking at any number in the table would allow you to get detailed information about the selected subset of jobs, like processing time stamps, exit code of application, Grid job id etc... If you sort your jobs by task and then click on a particular task name in the table, the task monitoring page for this task would be opened.

- Trying to understand the reason of the failures

If you see any discrepancy between information in the Dashboard and the output of the Crab status command

Click on the bar with the 'analysis' label and sort by site. Dark green colour corresponds to the jobs which finished properly, pink one corresponds to the jobs which were properly handled by the Grid, but failed from the application point of view. Red colour corresponds to the jobs aborted by the Grid. Clicking on the number corresponding to the failed or aborted jobs in the table below gives you the list of all failed or aborted jobs with their failure reason.

- Example:

Looking at the plot provided via link below you can see that there are no jobs which succeeded in Taipei

<https://twiki.cern.ch/twiki/pub/CMSPublic/WorkBookMonitoringTutorial/tut0.pdf>

Let's sort Taipei jobs by user

<https://twiki.cern.ch/twiki/pub/CMSPublic/WorkBookMonitoringTutorial/Tut1.pdf>

You see that there were several users running their jobs at the site and nobody managed to run the jobs properly. Clicking on the number of the failed jobs in the table we get the detailed view of the failed jobs with application exit code 8000, which very often indicates the data access problems (**beware** those images are from some time ago with older CMSSW releases, now failure to open file gives exit code=8020). So failures of the jobs could be related to the site misconfiguration rather than to a problem in the user code. If the problematic site does not represent the only location of data required by your task you can put the site in the black list (**ce_black_list=Site_Name**) of the Crab configuration file and resubmit the task. If you feel you have to do this black listing, also contact the crab support team, more information at the CRAB3 Troubleshooting page.

Review Status

Editor/Reviewer and date	Comments
NitishDhingra - 04-October-2017	Review, minor revisions
JohnStupak - 9-June-2013	Review, minor revisions
NitishDhingra - 07-Apr-2012	See detailed comments below
StefanoBelforte - 29-Jan-2010	Complete Expert Review, minor changes
Main.julia - 07 February 2008	author

Detailed comments 07-Apr-2012 [▶](#) [Hide](#) [▾](#)

Complete review with minor fixes. The page gives a very good illustration of the Dashboard monitoring .

Responsible: JuliaAndreeva

Last reviewed by: DaveEvans 28 Feb 2008

This topic: CMSPublic > WorkBookMonitoringTutorial

Topic revision: r32 - 2017-10-18 - NitishDhingra



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback