# Tutorial for CMS Users: Data Analysis on the Grid with CRAB

Benedikt Mura, Hartmut Stadie

Institut für Experimentalphysik, Universität Hamburg

September 2nd, 2009

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

# In this part you will learn . . .

1. how to locate CMS data
2. how to analyse CMS data on the grid (using CRAB)
3. how to store your job output on the grid
4. how to access this job output
   - by copying it to a local machine
   - interactively from the naf
   - with batch jobs on the naf
   - with grid jobs
5. how to monitor your grid jobs

# Outline

1. Locating Data
    - The Data Bookkeeping System
    - DBS Web Interface
    - DBS Query Language
2. The CMS Remote Analysis Builder (CRAB)
    - Introduction
    - Configuration
    - Running CRAB
    - CRAB Server
    - Storing the Output
    - Publishing the Output
    - CRAB on the NAF
3. Troubleshooting
    - Common Problems
4. Monitoring Grid Jobs
5. Dashboard

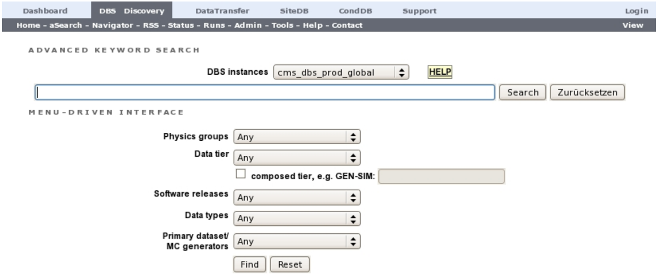
# Locating Data

The Data Bookkeeping System (DBS) is the authoritative source for information about stored data in CMS

- Information about produced data like location, file names, size, creation date, run number, . . .
- Information is stored in MySQL and Oracle data base backends
- Several instances for production, analysis, etc. Most important one: global DBS
- APIs allow communication with DBS used e.g. by Monte Carlo production tools
- A Query Language (QL) has been developped to support complex DBS queries. Those can be done via:
  1. Web interface: Data Discovery Service
  2. Command Line Interface

# DBS Web Interface

Webinterface for DBS searches:
`http://cmsweb.cern.ch/dbs_discovery/`



ADVANCED KEYWORD SEARCH

DBS instances: cms_dbs_prod_global

MENU-DRIVEN INTERFACE

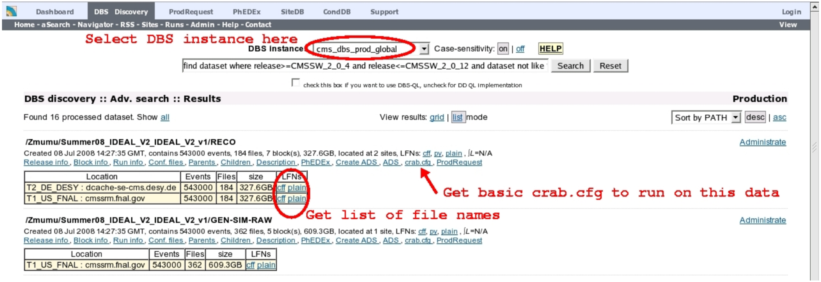
| | |
|---|---|
| Physics groups | Any |
| Data tier | Any |
| | □ composed tier, e.g. GEN-SIM: |
| Software releases | Any |
| Data types | Any |
| Primary dataset/ MC generators | Any |

- Menu driven search
- Query Language Search
- Links to help and examples
- Different instances selectable (in ’Production’ view only)

# DBS Web Interface

The search result gives access to information related to the dataset:

- Info about software releases, run numbers, lumi sections, related datasets, . . .
- Can obtain a basic configuration file to analyze this data with CRAB
- Can get a list of filenames (LFNs)



Select DBS instance here

DBS instance: cms_dbs_prod_global Case-sensitivity: on | off

find dataset where release>=CMSSW_2_0_4 and release<=CMSSW_2_0_12 and dataset not like

□ check this box if you want to use DBS-QL, uncheck for DD QL implementation

DBS discovery :: Adv. search :: Results — Production

Found 16 processed dataset. Show all — View results: grid | list mode — Sort by PATH

/Zmumu/Summer08_IDEAL_V2_IDEAL_V2_v1/RECO

Created 08 Jul 2008 14:27:35 GMT, contains 543000 events, 184 files, 7 block(s), 327.6GB, located at 2 sites, LFNs: cff, py, plain , ∫L=N/A

Release info , Block info , Run info , Conf. files , Parents , Children , Description , PhEDEx , Create ADS , ADS , crab.cfg , ProdRequest

| Location | Events | Files | size | LFNs |
|---|---|---|---|---|
| T2_DE_DESY : dcache-se-cms.desy.de | 543000 | 184 | 327.6GB | cff plain |
| T1_US_FNAL : cmssrm.fnal.gov | 543000 | 184 | 327.6GB | cff plain |

Get basic crab.cfg to run on this data

Get list of file names

/Zmumu/Summer08_IDEAL_V2_IDEAL_V2_v1/GEN-SIM-RAW

Created 08 Jul 2008 14:27:35 GMT, contains 543000 events, 362 files, 5 block(s), 609.3GB, located at 1 site, LFNs: cff, py, plain , ∫L=N/A

Release info , Block info , Run info , Conf. files , Parents , Children , Description , PhEDEx , Create ADS , ADS , crab.cfg , ProdRequest

| Location | Events | Files | size | LFNs |
|---|---|---|---|---|
| T1_US_FNAL : cmssrm.fnal.gov | 543000 | 362 | 609.3GB | cff plain |

Click around to find any information you need.

# DBS Query Language

The DBS query language is SQL syntax oriented and is rather human friendly:

- QL Syntax:
  find <key1>, <key2.attribute> where <key> <op> <value> and/or . . .
- Keywords: dataset, block, file, site, release, run, lumi, primds , procds, tier
- Operators: <=, <, >=, >, =, not like, like, in, between
- Some Attributes:
  - dataset: .name
  - block: .name, .size
  - file: .name,.numevents, .size
  - run: .number, numevents, .starttime, .endtime

Some documentation can be found on
`https://twiki.cern.ch/twiki/bin/view/CMS/QL`
and
`https://twiki.cern.ch/twiki/bin/view/CMS/DataDiscoveryInterface`

# DBS Query Language: Examples

Examples for the Data Discovery search

- *find dataset where dataset like **
  → look-up all datasets in DBS
- *find dataset where dataset like *Online* and site like T2_DE**
  → look-up datasets with dataset path matching pattern *Online* and located at a T2_DE site
- *find dataset where run between 34850 and 36000 or run in (34850,34890)*
  → look-up dataset with runs within given run ranges.
- *find file where release>CMSSW_1_6_7 and site=T2_UK*
  → find files located on T2_UK sites which were processed with release CMSSW_1_6_7 and above.
- *find file,run where dataset=/Commissioning2008Ecal-A/Online/RAW*
  → look-up files and runs for given dataset name
- *find sum(file.size),run where dataset=/Commissioning2008Ecal-A/Online/RAW*
  → look-up total size of files and list runs for given dataset name

A lot of examples can be found on: `https://twiki.cern.ch/twiki/bin/view/CMS/QL`

# CRAB

## What is CRAB?

The CMS Remote Analysis Builder (CRAB) is a command line tool for analysis of CMS data on the grid

Its purpose is

- to allow data analysis while hiding the complexities of the grid:
  - Find the distributed data
  - Take care of the runtime environment on the grid worker nodes
  - Execute the code the user developed locally
- to automate the workflow (status tracking, resubmissions, . . . ) (CRAB server)

Need to tell CRAB about the desired CMSSW version, the dataset and the CMSSW configuration file to use.

# CRAB Workflow

- Create task
  - Prepare jobs for remote environment
  - Split input data according to your request and its distribution among sites
- Submit the jobs
- Monitor job progress (CRAB server)
- Move job output to a storage element (or return it to the user)
- Resubmit failed jobs
- Notify about progress via email
- Publish the output (make it available for grid/CRAB jobs)

# CRAB Configuration

CRAB is steered with a single configuration file (`crab.cfg`):

- Parameters are set as `key = value` pairs
- Parameters are grouped in macro sections like `[CRAB]`, `[CMSSW]`, `[USER]`, `...`

Templates with a minimal/full parameterset are distributed with the CRAB release (`$CRABDIR/python/(full_)crab.cfg`)

# CRAB Configuration Example

A minimal `crab.cfg` file looks like this:

```
[CRAB]
jobtype = cmssw
scheduler = glite

[CMSSW]
### The data you want to access (to be found on DBS)
datasetpath=/RelValTTbar/CMSSW_3_1_0-MC_31X_V1-v1/GEN-SIM-RECO
### The ParameterSet you want to use
pset=PhysicsTools/PatAlgos/test/patLayer1_fromAOD_full_cfg.py
### Splitting parameters
total_number_of_events=-1
number_of_jobs = 15
### The output files (comma separated list)
output_file = PATLayer1_Output.fromAOD_full.root

[USER]
### OUTPUT files Management
return_data = 1

[GRID]
## RB/WMS management:
rb = CNAF
group = dcms
```

Will introduce more parameters later . . .

# CRAB: Before starting

First we need to set up the environment for gLite, CMSSW and CRAB:

- Set up the grid user interface: `ini glite`
- In your CMSSW `src`-directory: `cmsenv`
- Set up the local CRAB installation: `ini crab`

If you want to use the DESY WMS:

- edit your `crab.cfg` file:
  ```
  [GRID]
  rb = DESY
  ```
- copy its config in your working directory:
  ```
  cp $GLITE_LOCATION/etc/cms/glite_wms.conf
  glite_wms_DESY.conf
  ```

# CRAB: Basic commands I

After preparing the `crab.cfg` file you can start:

1. Create jobs:
   `crab -create`
   The project is created in a directory like `crab_0_080903_142652`
2. Submit jobs:
   `crab -submit [<all | number | range>] [-c <crab_dir>]`
   If `-c` is omitted the latest project in the current directory is taken.
3. Get status of jobs:
   `crab -status [-c <crab_dir>]`
4. Get output of finished jobs:
   `crab -getoutput [<all | job number | range>] [-c <crab_dir>]`
   The output is by default stored in `crab_dir/res/` and for each job *i* consists of `CMSSW_i.stdout, CMSSW_i.stderr, crab_fjr_i.xml` (and output_files).

# CRAB: Basic commands II

5. Cancel jobs:
   `crab -kill <job number/range> [-c <crab_dir>]`
6. Resubmit failed/cancelled jobs:
   `crab -resubmit <number | range> [-c <crab_dir>]`
7. Cleanup the directory
   `crab -clean [-c <crab_dir>]`

A descriptions of all available options can be obtained with `crab -h`
We will introduce a few more later .

# CRAB Server

The CRAB server

- monitores the progress of user jobs
- resubmits failed jobs
- notifies via email when a certain fraction has finished

The necessary configuration parameters are:

```
[CRAB]
server_name=bari
[USER]
eMail=your@email.address
## Fraction of finished jobs before email notification:
thresholdLevel=100
```

# CRAB: Copy output to a SE

Instead of retrieving your job output to the UI it is recommended to copy it on to storage element. In the `crab.cfg` this can be enabled with:

```
[USER]
return_data=0
copy_data=1
## Storage element at your home site or site name
storage_element= T2_DE_DESY
## subdirectory in your /store/user/<hypernews_name>
directory user_remote_dir = mySubDir
```

# Manipulating files on the SE

If you need to handle your files on a storage element

- dCache storage at DESY is mounted under `/pnfs/desy.de/cms/` and can be still explored using `ls`
- For managing your files on a storage element however you need to use grid tools
- You must setup the gLite UI and create a grid proxy before you can use them

Copy file from local disk to SE:

```
srmcp -2 "file:///path/file
srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/file"
```

Copy file from SE to local disk:

```
srmcp -2 -streams_num=1
"srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/file file:///path/file"
```

List directory:

```
srmls "srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/"
```

# More SRM Commands

Move/rename file:

```
srmmv "srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/oldfile
srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/newfile"
```

Remove file:

```
srmrm "srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/file"
```

If you want to remove all files from a directory:

```
for i in `ls /pnfs/desy.de/cms/tier2/store/user/user_name/directory` ; do
srmrm "srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/user_name/directory/$i" ; done
```

Make directory:

```
srmmkdir "srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/newdir"
```

Remove (empty) directory:

```
srmrmdir "srm://dcache-se-cms.desy.de:8443/srm/managerv2?SFN=
/pnfs/desy.de/cms/tier2/store/user/yourname/dir"
```

# Publishing your files in DBS I

Output files which you want to analyse further with CRAB jobs (e.g. a private MC production) have to be published in DBS.

```
[USER]
return_data=0
copy_data=1
storage_element= T2_DE_DESY
publish_data=1
publish_data_name=aNameForYourDataSet
# There are several instances, ask the expert in your institute which one
to use
dbs_url_for_publication=
https://cmsdbsprod.cern.ch:8443/cms_dbs_ph_analysis_02_writer/servlet/DBSServlet
publish_with_import_all_parents=0
```

The files will appear at:

```
SE+/store/user/<hypernewsname>/<PDName>/aNameForYourDataSet/<hash>
```

# Publishing your files in DBS II

Run CRAB as usual and retrieve the output. If all jobs have been successful do:

```
crab -publish
```

You can check the results on the command line:

```
InspectDBS2.py -DBSURL=<dbs_url> -datasetPath=<name_of_your_dataset>
```

or on the DBS web interface selecting the right instance in the production view

In order to run CRAB jobs on your own dataset you need to set:

```
[CMSSW]
dbs_url=
http://cmsdbsprod.cern.ch/cms_dbs_ph_analysis_02/servlet/DBSServlet
datasetpath=/OriginalDataSet/aNameForYourDataSet/USER
```

For details consult `https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrabForPublication`

You can delete your dataset from DBS with:

```
python DBSDeleteData.py -DBSURL=<dbs_url_for_adminstrator>
-datasetPath=<name_of_your_dataset>
```

# Documentation and Help

CRAB is documented on several webpages:

`https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookRunningGrid`

`https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrabHowTo`

`https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrabFaq`

Hypernews list for CRAB suppport:

`hn-cms-crabFeedback@cern.ch`

Most problems are very frequent, so seach the forum before writing

# CRAB on the NAF

CRAB can be used to submit jobs to the NAF batch system.

```
[CRAB]
scheduler=sge
[GRID]
# From the NAF you can only access data located at DESY
se_whitelist=desy.de
[SGE]
# parameters for SGE job submission
resource = -V -l h_vmem=2G -l site=hh -l h_cpu=0:60:00 -l os=’ls4|sl5’
```

Important: When using the local batch system you **cannot**:

- run on data located at other sites

# Common Problems

Troubleshooting:

- Strange python errors or commands not found: Environment not set up correctly. Set up gLite, CMSSW and CRAB in this order
- Failed jobs: After `crab -getoutput` do a `crab -status` to see the exit codes. Check the wiki for their meaning: `https://twiki.cern.ch/twiki//bin/view/CMS/JobExitCodes`
- Many failures are related to the job output:
  - 60302: output file not found → check file name to be identical in crab.cfg your CMSSW config file
  - 60303: file already exists in SE → Delete it or change the storage path
  - 70000: output too big → The size of the output sandbox is limited, use the `copy_data` option
- No compatible site found: Check your job requirements, availability of data and CMSSW version
- Get more information from the WMS about failed/aborted jobs: `crab -postMortem`

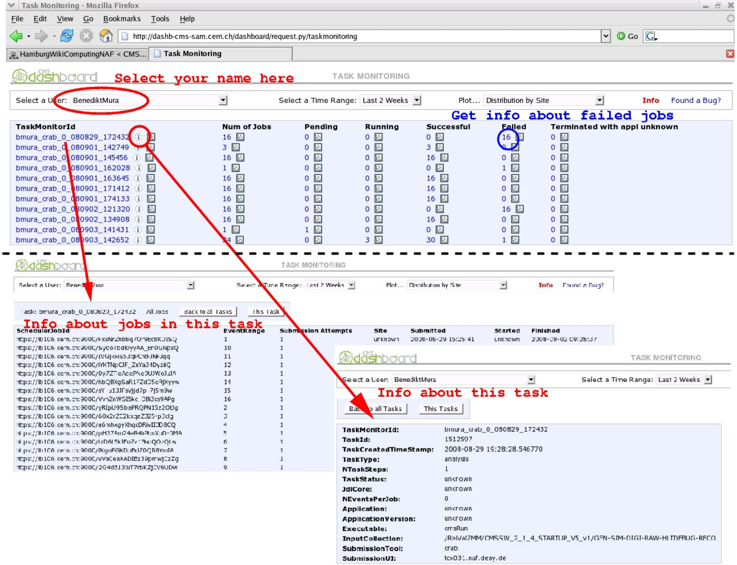
# Manual Job Control

One can manage CRAB jobs using the gLite tools:

- Get the job IDs for your crab jobs / the job collection
  - by `crab -printId`
  - by searching in `crab_dir/log/crab.log` for the Collection ID, e.g.: `https://lb106.cern.ch:9000/s22aPplJptmt6dsm2B9LWg`
- Use the glite commands to manage these jobs
  - glite-wms-job-status <jobID>
  - glite-wms-job-logging-info <jobID> (equivalent to -postMortem option)
  - glite-wms-job-output <jobID>
  - glite-wms-job-cancel <jobID>
- inspect JDL with `crab -createJdl 1`

# Monitoring Grid Jobs

Go to: `http://dashboard.cern.ch/cms/` → Task monitoring for the analysis users



Task Monitoring - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://dashb-cms-sam.cern.ch/dashboard/request.py/taskmonitoring
Go
HamburgWikiComputingNAF < CMS...
Task Monitoring
CMS dashboard
TASK MONITORING

Select your name here

Select a User: BenediktMura    Select a Time Range: Last 2 Weeks    Plot… Distribution by Site    Info    Found a Bug?

Get info about failed jobs

| TaskMonitorId | Num of Jobs | Pending | Running | Successful | Failed | Terminated with appl unknown |
|---|---|---|---|---|---|---|
| bmura_crab_0_080829_172432 | 16 | 0 | 0 | 0 | 16 | 0 |
| bmura_crab_0_080901_142749 | 3 | 0 | 0 | 3 | 0 | 0 |
| bmura_crab_0_080901_145456 | 16 | 0 | 0 | 16 | 0 | 0 |
| bmura_crab_0_080901_162028 | 1 | 0 | 0 | 0 | 1 | 0 |
| bmura_crab_0_080901_163645 | 16 | 0 | 0 | 16 | 0 | 0 |
| bmura_crab_0_080901_171412 | 16 | 0 | 0 | 16 | 0 | 0 |
| bmura_crab_0_080901_174133 | 16 | 0 | 0 | 16 | 0 | 0 |
| bmura_crab_0_080902_121320 | 16 | 0 | 0 | 0 | 16 | 0 |
| bmura_crab_0_080902_134908 | 16 | 0 | 0 | 16 | 0 | 0 |
| bmura_crab_0_080903_141431 | 1 | 1 | 0 | 0 | 0 | 0 |
| bmura_crab_0_080903_142652 | 34 | 0 | 3 | 30 | 1 | 0 |

Select a User: BenediktMura    Select a Time Range: Last 2 Weeks    Plot… Distribution by Site    Info    Found a Bug?

Task: bmura_crab_0_080829_172432    All Jobs    Back to all Tasks    This Task

Info about jobs in this task

| SchedulerJobId | EventRange | Submission Attempts | Site | Submitted | Started | Finished |
|---|---|---|---|---|---|---|
| https://lb106.cern.ch:9000/FxsNrZhb8q7O-9Bc8KJISQ | 1 | 1 | unknown | 2008-08-29 15:25:41 | unknown | 2008-09-02 09:28:37 |
| https://lb106.cern.ch:9000/SycoTbdI0yyAA_Br0GNpsQ | 10 | 1 | | | | |
| https://lb106.cern.ch:9000/1VGjkxiS53gPCntHJNFJqq | 11 | 1 | | | | |
| https://lb106.cern.ch:9000/IYKTNpCJF_ZxYa34DyziiQ | 12 | 1 | | | | |
| https://lb106.cern.ch:9000/0y7ZTeAdcPNeDUJWoJJA | 13 | 1 | | | | |
| https://lb106.cern.ch:9000/hbQBXgSaR1FZdD5cRjXyyw | 14 | 1 | | | | |
| https://lb106.cern.ch:9000/sY u13JFsVjjd7p-7j5m9w | 15 | 1 | | | | |
| https://lb106.cern.ch:9000/VvnZnWSI5kc DBi3cs9APg | 16 | 1 | | | | |
| https://lb106.cern.ch:9000/yRIpU9SbePRQPN15z2OtJg | 2 | 1 | | | | |
| https://lb106.cern.ch:9000/60x2rZC2kxqeZ325-p3clg | 3 | 1 | | | | |
| https://lb106.cern.ch:9000/a6mhxgyXhqcDRivII3D8CQ | 4 | 1 | | | | |
| https://lb106.cern.ch:9000/pH37Am24eR4hPtnXpD-0MA | 5 | 1 | | | | |
| https://lb106.cern.ch:9000/LrD6L5kIFoZe1RxQO/QLw | 6 | 1 | | | | |
| https://lb106.cern.ch:9000/fKgpFGKDpBdZ0QTi8YmTA | 7 | 1 | | | | |
| https://lb106.cern.ch:9000/uVaCeaXADIBz39pmwjCzZg | 8 | 1 | | | | |
| https://lb106.cern.ch:9000/2G4d313tuT7rbKZjCV6UDw | 9 | 1 | | | | |

Select a User: BenediktMura    Select a Time Range: Last 2 Weeks

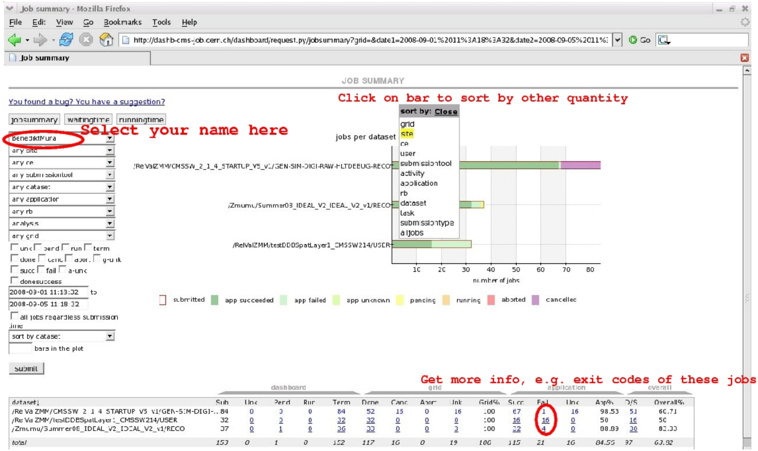
Info about this task

Back to all Tasks    This Tasks

| | |
|---|---|
| TaskMonitorId: | bmura_crab_0_080829_172432 |
| TaskId: | 1512597 |
| TaskCreatedTimeStamp: | 2008-08-29 19:28:28.546770 |
| TaskType: | analysis |
| NTaskSteps: | 1 |
| TaskStatus: | unknown |
| JdlCore: | unknown |
| NEventsPerJob: | 0 |
| Application: | unknown |
| ApplicationVersion: | unknown |
| Executable: | cmsRun |
| InputCollection: | /RelValZMM/CMSSW_2_1_4_STARTUP_V5_v1/GEN-SIM-DIGI-RAW-HLTDEBUG-RECO |
| SubmissionTool: | crab |
| SubmissionUI: | tcx031.naf.desy.de |

# Monitoring Grid Jobs

Go to: `http://dashboard.cern.ch/cms/` → Interactive View



Job summary - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://dashb-cms-job.cern.ch/dashboard/request.py/jobsummary?grid=&date1=2008-09-01%2011%3A18%3A32&date2=2008-09-05%2011%

JOB SUMMARY

You found a bug? You have a suggestion?

jobsummary · waitingtime · runningtime

Select your name here

- BenediktMura
- any site
- any ce
- any submissiontool
- any dataset
- any application
- any rb
- analysis
- any grid

□ unk □ pend □ run □ term
□ done □ canc □ abort □ g-unk
□ succ □ fail □ a-unk
□ donesuccess

2008-09-01 11:18:32 to
2008-09-05 11:18:32

□ all jobs regardless submission time

sort by dataset

bars in the plot

submit

Click on bar to sort by other quantity

sort by: Close

- grid
- site
- ce
- user
- submissiontool
- activity
- application
- rb
- dataset
- task
- submissiontype
- alljobs

jobs per dataset
/RelValZMM/CMSSW_2_1_4_STARTUP_V5_v1/GEN-SIM-DIGI-RAW-HLTDEBUG-RECO
/Zmumu/Summer08_IDEAL_V2_IDEAL_V2_v1/RECO
/RelValZMM/testDDDSpatLayer1_CMSSW214/USER
10
20
30
40
50
60
70
80
number of jobs

□ submitted 🟩 app succeeded □ app failed □ app unknown □ pending □ running 🟥 aborted □ cancelled

Get more info, e.g. exit codes of these jobs

| | dashboard | | | | grid | | | | | application | | | | overall | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| dataset↓ | Sub | Unk | Pend | Run | Term | Done | Canc | Abort | Unk | Grid% | Succ | Fail | Unk | App% | O/S | Overall% |
| /RelValZMM/CMSSW_2_1_4_STARTUP_V5_v1/GEN-SIM-DIGI-... | 84 | 0 | 0 | 0 | 84 | 52 | 15 | 0 | 16 | 100 | 67 | 1 | 16 | 98.53 | 51 | 60.71 |
| /RelValZMM/testDDDSpatLayer1_CMSSW214/USER | 32 | 0 | 0 | 0 | 32 | 32 | 0 | 0 | 0 | 100 | 16 | 16 | 0 | 50 | 16 | 50 |
| /Zmumu/Summer08_IDEAL_V2_IDEAL_V2_v1/RECO | 37 | 0 | 1 | 0 | 36 | 33 | 0 | 0 | 3 | 100 | 32 | 4 | 0 | 88.89 | 30 | 83.33 |
| total | 153 | 0 | 1 | 0 | 152 | 117 | 16 | 0 | 19 | 100 | 115 | 21 | 16 | 84.55 | 97 | 63.92 |

# Monitoring Grid Jobs

- There is a delay for your jobs to show up
- Info on dashboard may differ from CRAB/glite job info. The later is more reliable, because reporting to the dashboard might fail.
- If many of your jobs fail check if this is a site problem:
  - Check if they all ran at the same site and if other user get the same errors at that site
  - Check the ’Site status board’ link on the dashboard: check the ’Analysis’ column
  - If you have problems at a particular site try to use another one hosting that dataset (black/white list in crab.cfg)
  - If the problem persists inform site admins/open a GGUS ticket