Table of Contents

5.6.1 Running CMSSW code on the Grid using CRAB2

(for CRAB3 tutorial please click HERE)

Complete:
Detailed Review status

WARNING

- **You should always use latest production CRAB version**
- This tutorial **is** outdated since it was prepared for a live lesson at a specific time and thus refers to a particular dataset and CMSSW version that may not be available when you read this (and where you try it).
    - **as of 2014 you should be able to kickstart your Crab work using CMSSW 5_3_11 and the dataset /GenericTTbar/HC-CMSSW_5_3_1_START53_V5-v1/GEN-SIM-RECO as MC data and /SingleMu/Run2012B-13Jul2012-v1/AOD as real data.**

Contents:

Prerequisites to run the tutorial

- to have a valid Grid certificate
- to be registered to the CMS virtual organization
    - **to get the Grid certificate and to register to VO CMS please follow the CRAB howto instructions**
- to be registered to the siteDB
    - **please follow the instruction at siteDB registration for CRAB**
- to have access to lxplus machines or to an SLC5 User Interface

Recipe for the tutorial

For this tutorial we will refer to CMS software:

- **CMSSW_5_3_11**

and we will use an already prepared CMSSW analysis code to analyze the sample:

- The tutorial will focus on the basic workflow using the dataset: */RelValProdTTbar/JobRobot-MC_3XY_V24_JobRobot-v1/GEN-SIM-DIGI-RECO* (MC dataset) and */SingleMu/Run2012B-13Jul2012-v1/AOD* (real data): CRAB configuration file for real data with lumi mask

We will use the central installation of CRAB available at CERN:

- **CRAB_2_9_1**

The example is written to use the *csh* shell family. If you want to use the Bourne Shell replace *csh* with *sh*.

Legend of colors for this tutorial

```
BEIGE background for the commands to execute  (cut&paste)

GREEN background for the output sample of the executed commands (nearly what you should see in yo

BLUE background for the configuration files  (cut&paste)
```

Setup local Environment and prepare user analysis code

In order to submit jobs to the Grid, you **must** have an access to a LCG User Interface (LCG UI). It will allow you to access WLCG-affiliated resources in a fully transparent way. LXPLUS users can get an LCG UI via AFS by:

```
source /afs/cern.ch/cms/LCG/LCG-2/UI/cms_ui_env.csh
```

Install CMSSW project in a directory of your choice. In this case we create a "TESTfirst " directory:

```
mkdir TEST
cd TEST
cmsrel CMSSW_5_3_11
#cmsrel is an alias of scramv1 project CMSSW CMSSW_5_3_11
cd CMSSW_5_3_11/src/
cmsenv
#cmsenv is an alias for scramv1 runtime -csh
```

For this tutorial we are going to use as CMSSW configuration file, the tutorial.py:

```
import FWCore.ParameterSet.Config as cms
process = cms.Process('Slurp')

process.source = cms.Source("PoolSource", fileNames = cms.untracked.vstring())
process.maxEvents = cms.untracked.PSet( input       = cms.untracked.int32(10) )
process.options   = cms.untracked.PSet( wantSummary = cms.untracked.bool(True) )

process.output = cms.OutputModule("PoolOutputModule",
    outputCommands = cms.untracked.vstring("drop *", "keep recoTracks_*_*_*"),
    fileName = cms.untracked.string('outfile.root'),
)
process.out_step = cms.EndPath(process.output)
```

CRAB setup

Setup on lxplus:

In order to setup and use CRAB from any directory, source the script `crab.(c)sh` located in `/afs/cern.ch/cms/ccs/wm/scripts/Crab/`, which always points to the latest version of CRAB. After the source of the script it's possible to use CRAB from any directory (typically use it on your CMSSW working directory).

```
source /afs/cern.ch/cms/ccs/wm/scripts/Crab/crab.csh
```

Warning: in order to have the correct environment, the order to source env files has always to be

- source of UI env
- setup of CMSSW software

- source of CRAB env

Locate the dataset and prepare CRAB submission

In order to run our analysis over a whole dataset, we have to find first the data name and then put it on the `crab.cfg` configuration file.

Data selection

To select data you want to access, use the **DAS** web page where available datasets are listed Data Aggregation Service (DAS) . For this tutorial we'll use :

```
/RelValProdTTbar/JobRobot-MC_3XY_V24_JobRobot-v1/GEN-SIM-DIGI-RECO
 (MC data)
```

- Beware: datasets availability as sites changes with time, if you are trying to follow this tutorial after the date it was given, you may need to use another one

CRAB configuration

Modify the CRAB configuration file `crab.cfg` according to your needs: a fully documented template is available at `$CRABPATH/full_crab.cfg`, a template with essential parameters is available at `$CRABPATH/crab.cfg`. The default name of configuration file is `crab.cfg`, but you can rename it as you want.

Copy one of these files in your local area.

For guidance, see the list and description of configuration parameters in the on-line CRAB manual. For this tutorial, the only relevant sections of the file are `[CRAB]`, `[CMSSW]` and `[USER]` .

Configuration parameters

The list of the main parameters you need to specify on your `crab.cfg`:

- **pset**: the CMSSW configuration file name;
- **output_file**: the output file name produced by your pset; if in the CMSSW pset the output is defined in TFileService, the file is automatically handled by CRAB, and there is no need to specify it on this parameter;
- **datasetpath**: the full dataset name you want to analyze;
- ***Jobs splitting***:
  - By event: only for MC data. You need to specify 2 of these parameters: **total_number_of_events**, **number_of_jobs**, **events_per_job**
    - specify the *total_number_of_events* and the *number_of_jobs*: this will assign to each job a number of events equal to *total_number_of_events/number_of_jobs*
    - specify the *total_number_of_events* and the *events_per_job*: this will assign to each job *events_per_job* events and will calculate the number of jobs by *total_number_of_events/events_per_job*;
    - or you can specify the *number_of_jobs* and the *events_per_job*;
  - By lumi: real data require it. You need to specify 2 of these parameters: **total_number_of_lumis**, **lumis_per_job**, **number_of_jobs**

    - ◊ because jobs in split-by-lumi mode process entire rather than partial files, you will often end up with fewer jobs processing more lumis than expected. Additionally, a single job cannot analyze files from multiple blocks in DBS. So these parameters are "advice" to CRAB rather than determinative.
    - ◊ specify the *lumis_per_job* and the *number_of_jobs*: the total number of lumis processed will be *number_of_jobs* x *lumis_per_job*
    - ◊ or you can specify the *total_number_of_lumis* and the *number_of_jobs*
    - ◊ **lumi_mask**: the filename of a JSON file that describes which runs and lumis to process. CRAB will skip luminosity blocks not listed in the file.
- **return_data**: this can be 0 or 1; if it is one you will retrieve your output files to your local working area;
- **copy_data**: this can be 0 or 1; if it is one you will copy your output files to a remote Storage Element;
- **local_stage_out**: this can be 0 or 1; if this is one your produced output is copied to the closeSE in the case of failure of the copy to the SE specified in your crab.cfg
- **publish_data**: this can be 0 or 1; if it is one you can publish your produced data to a local DBS;
- **scheduler**: the name of the scheduler you want to use;
- **jobtype**: the type of the jobs.

Run CRAB on MonteCarlo data copying the output to a Storage Element

The chance to copy the output to an existing **Storage Element** allows to bypass the output size limit constraint, to publish the data on a local DBS and then to easily re-run over the published data. In order to make CRAB copies to a Storage Element you have to add the following information on the Crab configuration file:

- that we want to copy our results adding **copy_data=1** and **return_data=0** (it is not allowed to have both at 1);
- add the **official CMS site name** where we are going to copy our results; the name of official CMS sites can be found in the siteDB

CRAB configuration file for MonteCarlo data

You can find more details on this at the corresponding link on the CRAB FAQ page.

The CRAB configuration file (default name crab.cfg) should be located at the same location as the CMSSW parameter-set to be used by CRAB with the following content:

```
[CMSSW]
total_number_of_events  = 10
number_of_jobs          = 5
pset                    = tutorial.py
datasetpath             =  /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO

output_file              = outfile.root

[USER]
return_data             = 0
copy_data               = 1
storage_element        = T2_xx_yyyy (to change with the CMS name of site where you can write outp
user_remote_dir         = TutGridSchool

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

Run Crab

Once your `crab.cfg` is ready and the whole underlying environment is set up, you can start running CRAB. CRAB supports command line help which can be useful for the first time. You can get it via:

```
crab -h
```

Job Creation

The job creation checks the availability of the selected dataset and prepares **all** the jobs for submission according to the selected job splitting specified in the crab.cfg

- By default the creation process creates a CRAB project directory (default: crab_0_date_time) in the current working directory, where the related crab configuration file is cached for further usage, avoiding interference with other (already created) projects
- Using the [USER] *ui_working_dir* parameter in the configuration file CRAB allows the user to chose the project name, so that it can be used later to distinguish multiple CRAB projects in the same directory.

```
crab -create
```

that takes by default the configuration file called crab.cfg associated for this tutorial with MC data.

The creation command could ask for proxy/myproxy passwords the first time you use it and it should produce a similar screen output like:

```
$ crab -create
crab:  Version 2.9.1 running on Fri Oct 11 15:33:18 2013 CET (13:33:18 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Contacting Data Discovery Services ...
crab:  Accessing DBS at: http://cmsdbsprod.cern.ch/cms_dbs_prod_global/servlet/DBSServlet
crab:  Requested dataset: /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO has 9513 events in 1

crab:  SE black list applied to data location: ['srm-cms.cern.ch', 'srm-cms.gridpp.rl.ac.uk', 'T1
crab:  May not create the exact number_of_jobs requested.
crab:  5 job(s) can run on 10 events.

crab:  List of jobs and available destination sites:

Block     1: jobs                  1-5: sites: T2_CH_CERN, T1_US_FNAL_MSS

crab:  Checking remote location
crab:  Creating 5 jobs, please wait...
crab:  Total of 5 jobs created.

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/log/crab.log
```

* the project directory called crab_0_131011_153317 is created

Job Submission

With the submission command it's possible to specify a combination of jobs and job-ranges separated by comma (e.g.: =1,2,3-4), the default is all. To submit all jobs of the last created project with the default name, it's enough to execute the following command:

```
crab -submit
```

to submit a specific project:

```
crab -submit -c  <dir name>
```

which should produce a similar screen output like:

```
$ crab -submit
crab:  Version 2.9.1 running on Fri Oct 11 15:33:34 2013 CET (13:33:34 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Checking available resources...
crab:  Found  compatible site(s) for job 1
crab:  1 blocks of jobs will be submitted
crab:  remotehost from Avail.List = vocms83.cern.ch
crab:  contacting remote host vocms83.cern.ch
crab:  Establishing gsissh ControlPath. Wait 2 sec ...
crab:  Establishing gsissh ControlPath. Wait 2 sec ...
crab:  Establishing gsissh ControlPath. Wait 2 sec ...
crab:  COPY FILES TO REMOTE HOST
crab:  SUBMIT TO REMOTE GLIDEIN FRONTEND
                                                                          Submitting 5 jobs
100% [=============================================================================================
                                                                            please wait
Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/log/crab.log
```

Job Status Check

Check the status of the jobs in the latest CRAB project with the following command:

```
crab -status
```

to check a specific project:

```
crab -status -c  <dir name>
```

which should produce a similar screen output like:

```
$ crab -status
crab:  Version 2.9.1 running on Fri Oct 11 15:42:49 2013 CET (13:42:49 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/
```

```
crab:  error detecting glite version
crab:  error detecting glite version
crab:  Checking the status of all jobs: please wait
crab:  contacting remote host vocms83.cern.ch
crab:
ID    END STATUS            ACTION       ExeExitCode JobExitCode E_HOST
----- --- ----------------- ------------  ---------- ----------- ---------
1     N   Running           SubSuccess                            cmsosgce.fnal.gov
2     N   Running           SubSuccess                            cmsosgce.fnal.gov
3     N   Running           SubSuccess                            cmsosgce.fnal.gov
4     N   Running           SubSuccess                            cmsosgce.fnal.gov
5     N   Running           SubSuccess                            cmsosgce.fnal.gov

crab:   5 Total Jobs
 >>>>>>>>> 5 Jobs Running
        List of jobs Running: 1-5

crab:  You can also follow the status of this task on :
        CMS Dashboard: http://dashb-cms-job-task.cern.ch/taskmon.html#task=fanzago_crab_0_131011_
        Your task name is: fanzago_crab_0_131011_153317_hg41w0

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/log/crab.log
```

Job Output Retrieval

For the jobs which are in the "Done" status it is possible to retrieve the log files of the jobs (just the log files, because the output files are copied to the Storage Element associated to the T2 specified on the crab.cfg and infact return_data is 0). The following command retrieves the log files of all "Done" jobs of the last created CRAB project:

```
crab -getoutput
```

to get the output of a specific project:

```
crab -getoutput -c  <dir name>
```

the job results (CMSSW_n.stdout, CMSSW_n.stderr and crab_fjr_n.xml) will be copied in the `res` subdirectory of your crab project:

```
$ crab -get
crab:  Version 2.9.1 running on Fri Oct 11 16:17:23 2013 CET (14:17:23 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  contacting remote host vocms83.cern.ch
crab:  Preparing to rsync 2 files
crab:  Results of Jobs # 1 are in /afs/cern.ch/user/f/fanzago/scratch0/TEST_RELEASE/TEST_PATC2/TE
crab:  contacting remote host vocms83.cern.ch
crab:  Preparing to rsync 8 files
crab:  Results of Jobs # 2 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153
crab:  Results of Jobs # 3 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153
crab:  Results of Jobs # 4 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153
crab:  Results of Jobs # 5 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153
Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/crab_0_131011_1533
```

The stderr is an empty file, the stdout is the output of the wrapper of your analysis code (the output of CMSSW.sh script created by CRAB) and the crab_fjr.xml is the FrameworkJobReport created by your analysis code.

Use the -report option

Print a short report about the task, namely the total number of events and files processed/requested/available, the name of the dataset path, a summary of the status of the jobs, and so on. A summary file of the runs and luminosity sections processed is written to res/. In principle -report should generate all the info needed for an analysis. Command to execute:

```
crab -report
```

Example of execution:

```
$ crab -report
crab:  Version 2.9.1 running on Fri Oct 11 17:02:17 2013 CET (15:02:17 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  --------------------
Dataset: /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO
Remote output :
SE: T2_CH_CERN srm-eoscms.cern.ch  srmPath: srm://srm-eoscms.cern.ch:8443/srm/v2/server?SFN=/eos/
Total Events read: 10
Total Files read: 5
Total Jobs : 5
Luminosity section summary file: /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_1533
   # Jobs: Retrieved:5

---------------------------

crab:  The summary file inputLumiSummaryOfTask.json about input run and lumi isn't created
crab:  No json file to compare
```

The message "The summary file inputLumiSummaryOfTask.json about input run and lumi isn't created" isn't an error but a message that means input data didn't provide lumi section info, as expected for the MC data.

The full srm path will allow you to know where your data has been stored and to perform operations by hand on it. As example you can delete the data using **srmrm** command and check the content of the remote directory through **srmls**. In this case the remote directory is:

```
srm://srm-eoscms.cern.ch:8443/srm/v2/server?SFN=/eos/cms/store/user/fanzago/TutGridSchool_test
```

It could be necessary to substitute the ? with the "?" in the srm path, depending on the shell you are using. Additional srm commands include `srmrm`, `srmrmdir`, `srmmv`, for moving files within an srm system, `srmcp` which can copy files locally. Note that to copy files locally, `srmcp` may require the additional flag "-2" to ensure that the version 2 client is used.

Here is the content of the file containing the luminosity summary */crab_0_130220_173930/res/lumiSummary.json*:

```
{"1": [[666666, 666666]]}
```

Copy the output from the SE to the local User Interface

Option that can be used only if your output have been previously copied by CRAB on a remote SE. By default the -copyData copies your output from the remote SE to the local CRAB working directory (under res). Otherwise you can copy the output from the remote SE to another one, specifying either -dest_se= or -dest_endpoint=. If dest_se is used, CRAB finds the correct path where the output can be stored. The command to execute in order to retrieve locally the remote output files to your local user interface is:

```
crab -copyData
## or crab -copyData -c <dir name>
```

An example of execution:

```
$ crab -copyData
crab:  Version 2.9.1 running on Fri Oct 11 17:08:38 2013 CET (15:08:38 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Copy file locally.
        Output dir: /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/res/
crab:  Starting copy...
directory/afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/res/already exists
crab:  Copy success for file: outfile_4_1_Jlr.root
crab:  Copy success for file: outfile_3_1_MsR.root
crab:  Copy success for file: outfile_1_1_HF3.root
crab:  Copy success for file: outfile_2_1_cVA.root
crab:  Copy success for file: outfile_5_1_gAw.root
Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131011_153317/log/crab.log
```

Publish your result in DBS

The publication of the produced data to DBS allows to re-run over the produced data that has been published. The instructions to follow are below, and here is the link to the how to. You have to add to the Crab configuration file more information specifying that you (will) want to publish and the data name to publish.

```
[USER]
....
publish_data            = 1
publish_data_name       = what_you_want
....
```

Warning:

- All the parameters related publication have to be added in the configuration file before creation of jobs, even if the publication step is executed after retrieving of job output.
- Publication is done in the phys03 instance of DBS3. If you belong to a PAG group, you have to publish your data to the DBS associated to your group, checking at the DBS access twiki page the correct DBS url and which role in voms you need to be an allowed user.

- Remember to change the *ui_working_dir* value in the configuration file to create a new project (if you don't use the default name of crab project), otherwise the creation step will fail with the error message "project already exists, please remove it before create new task ".

Run Crab publishing your results

You can also run your analysis code publishing the results copied to a remote Storage Element. Here below an example of the CRAB configuration file, coherent with this tutorial:

For MC data (crab.cfg)

```
[CMSSW]
total_number_of_events  = 50
number_of_jobs          = 10
pset                    = tutorial.py
datasetpath             = /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO
output_file              = outfile.root

[USER]
return_data             = 0
copy_data               = 1
storage_element         = T2_xx_yyyy
publish_data            = 1
publish_data_name       = FanzagoTutGrid

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

And with this `crab.cfg` you can re-do the complete workflow as described before, plus the publication step:

- creation
- submission
- status progress monitoring
- output retrieval
- publish the results

Use the -publish option

After having done the previous workflow untill the retrieval of you jobs, you can publish the output data that have been stored in the Storage Element indicated in the `crab.cfg` file using:

```
crab -publish
```

or to publish the outputs of a specific project:

```
crab -publish -c <dir_name>
```

It is not necessary that all the jobs are done and retrieved. You can publish your output at a different time.

It will look for all the FrameworkJobReport files (crab-project-dir/res/crab_fjr_*.xml) produced by each job and will extract from there the information (i.e. number of events, LFN, etc.) to publish.

Publication output example

The output shown below corresponds to an old output using DBS2.

```
$ crab -publish
crab:  Version 2.9.1 running on Mon Oct 14 14:35:56 2013 CET (12:35:56 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_123645/

crab:  <dbs_url_for_publication> = https://cmsdbsprod.cern.ch:8443/cms_dbs_ph_analysis_02_writer/
file_list =  ['/afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_123645/res//crab_fjr_1

crab:  --->>> Start dataset publication
crab:  --->>> Importing parent dataset in the dbs: /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-
crab:  --->>> Importing all parents level
--------------------------------------------------------------------------------
Transferring path /RelValZMM/CMSSW_5_2_1-START52_V4-v1/GEN-SIM
           block /RelValZMM/CMSSW_5_2_1-START52_V4-v1/GEN-SIM#24e1effb-0f0c-4557-bb46-3d5ecae691b
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Transferring path /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-DIGI-RAW-HLTDEBUG
            block /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-DIGI-RAW-HLTDEBUG#13e93136-29ed-1
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Transferring path /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO
            block /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO#43683124-29f6-11e2-9c63-0022
--------------------------------------------------------------------------------

crab:  --->>> duration of all parents import (sec): 552.62570405
crab:  Import ok of dataset /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO
crab:  PrimaryDataset = RelValZMM
crab:  ProcessedDataset = fanzago-FanzagoTutGrid-f30a6bb13f516198b2814e83414acca1
crab:  <User Dataset Name> = /RelValZMM/fanzago-FanzagoTutGrid-f30a6bb13f516198b2814e83414acca1/U

crab:  --->>> End dataset publication
crab:  --->>> Start files publication
crab:  --->>> End files publication
crab:  --->>> Check data publication: dataset /RelValZMM/fanzago-FanzagoTutGrid-f30a6bb13f516198b

=== dataset /RelValZMM/fanzago-FanzagoTutGrid-f30a6bb13f516198b2814e83414acca1/USER
=== dataset description =
===== File block name: /RelValZMM/fanzago-FanzagoTutGrid-f30a6bb13f516198b2814e83414acca1/USER#78
      File block located at:  ['t2-srm-02.lnl.infn.it']
      File block status: 0
      Number of files: 10
      Number of Bytes: 33667525
      Number of Events: 50

 total events: 50 in dataset: /RelValZMM/fanzago-FanzagoTutGrid-f30a6bb13f516198b2814e83414acca1/

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_123645/log/crab.log
```

Warning: Some versions of CMSSW switch off the debug mode of crab, so a lot of duplicated info can be reported at screen level.

Analyze your published data

First note that:

- CRAB by default publishes all files finished correctly, including files with 0 events
- CRAB by default imports all dataset parents of your dataset

You have to modify your `crab.cfg` file specifying the datasetpath name of your dataset and the dbs_url where data are published (we will assume phys03 instance of DBS3):

```
[CMSSW]
....
datasetpath = your_dataset_path
dbs_url = phys03
```

The creation output will be something similar to:

```
$ crab -create
crab:  Version 2.9.1 running on Mon Oct 14 15:49:31 2013 CET (13:49:31 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_154931/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Contacting Data Discovery Services ...
crab:  Accessing DBS at: https://cmsweb.cern.ch/dbs/prod/phys03/DBSReader
crab:  Requested dataset: /RelValZMM/fanzago-FanzagoTutGrid-f30a6bb13f516198b2814e83414acca1/USER

crab:  SE black list applied to data location: ['srm-cms.cern.ch', 'srm-cms.gridpp.rl.ac.uk', 'T1
crab:  May not create the exact number_of_jobs requested.
crab:  10 job(s) can run on 50 events.

crab:  List of jobs and available destination sites:

Block     1: jobs                1-10: sites: T2_IT_Legnaro

crab:  Checking remote location
crab:  WARNING: The stageout directory already exists. Be careful not to accidentally mix outputs
crab:  Creating 10 jobs, please wait...
crab:  Total of 10 jobs created.

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_154931/log/crab.log
```

The jobs will run in the site where your USER data have been stored.

CRAB configuration file for real data with lumi mask

You can find more details on this at the corresponding link on the Crab FAQ page.

The CRAB configuration file (default name crab.cfg) should be located at the same location as the CMSSW parameter-set to be used by CRAB. The dataset used is: */SingleMu/Run2012B-13Jul2012-v1/AOD*

For real data (crab_lumi.cfg)

```
[CMSSW]
lumis_per_job           = 50
number_of_jobs          = 10
```

```
pset                    = tutorial.py
datasetpath             = /SingleMu/Run2012B-13Jul2012-v1/AOD
lumi_mask              = Cert_190456-208686_8TeV_PromptReco_Collisions12_JSON.txt
output_file             = outfile.root

[USER]
return_data              = 0
copy_data                = 1
publish_data             = 1
publish_data_name       = FanzagoTutGrid_data

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

where the lumi_mask file can be downloaded with

```
wget --no-check-certificate https://cms-service-dqm.web.cern.ch/cms-service-dqm/CAF/certification
```

For the tutorial we are using a subset of run and lumi (using a lumiMask.json file). The lumi_mask file (Cert_190456-208686_8TeV_PromptReco_Collisions12_JSON.txt) contains:

```
{"190645": [[10, 110]], "190704": [[1, 3]], "190705": [[1, 5], [7, 76], [78, 336], [338, 350], [3
...
"208551": [[119, 193], [195, 212], [215, 300], [303, 354], [356, 554], [557, 580]], "208686": [[7
```

Job Creation

Creating jobs for real data is analogous to montecarlo data. To not overwrite previous run for this tutorial, it is suggested to use a dedicated cfg:

```
crab -create -cfg crab_lumi.cfg
```

that takes as configuration file the file name specified with the option -cfg, in this case the crab_lumi.cfg associated for this tutorial with real data.

```
$ crab -create -cfg crab_lumi.cfg
crab:  Version 2.9.1 running on Mon Oct 14 16:05:18 2013 CET (14:05:18 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Contacting Data Discovery Services ...
crab:  Accessing DBS at: http://cmsdbsprod.cern.ch/cms_dbs_prod_global/servlet/DBSServlet
crab:  Requested (A)DS /SingleMu/Run2012B-13Jul2012-v1/AOD has 14 block(s).
crab:  SE black list applied to data location: ['srm-cms.cern.ch', 'srm-cms.gridpp.rl.ac.uk', 'T1
crab:  Requested number of lumis reached.
crab:  9 jobs created to run on 500 lumis
crab:  Checking remote location
crab:  Creating 9 jobs, please wait...
crab:  Total of 9 jobs created.
Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/log/crab.log
```

- The project directory called crab_0_131014_160518 is created.
- As explained the number of created jobs can not match the number of jobs required in the configuration file (9 created but 10 required jobs).

Job Submission

Job submission is always analogous:

```
$ crab -submit
crab:  Version 2.9.1 running on Mon Oct 14 16:07:59 2013 CET (14:07:59 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Checking available resources...
crab:  Found  compatible site(s) for job 1
crab:  1 blocks of jobs will be submitted
crab:  remotehost from Avail.List = submit-4.t2.ucsd.edu
crab:  contacting remote host submit-4.t2.ucsd.edu
crab:  Establishing gsissh ControlPath. Wait 2 sec ...
crab:  Establishing gsissh ControlPath. Wait 2 sec ...
crab:  COPY FILES TO REMOTE HOST
crab:  SUBMIT TO REMOTE GLIDEIN FRONTEND
                                                                                Submitting 9 jobs
100% [=============================================================================================
                                                                                  please wait
Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/log/crab.log
```

Job Status Check

Check the status of the jobs in the latest CRAB project with the following command:

```
crab -status
```

to check a specific project:

```
crab -status -c  <dir name>
```

which should produce a similar screen output like:

```
[fanzago@lxplus0445 SLC6]$ crab -status
crab:  Version 2.9.1 running on Mon Oct 14 16:23:52 2013 CET (14:23:52 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Checking the status of all jobs: please wait
crab:  contacting remote host submit-4.t2.ucsd.edu
crab:
ID    END STATUS            ACTION       ExeExitCode JobExitCode E_HOST
----- --- ----------------- ------------  ---------- ----------- ---------
1     N   Running           SubSuccess                            ce208.cern.ch
2     N   Submitted         SubSuccess
3     N   Running           SubSuccess                            cream03.lcg.cscs.ch
4     N   Running           SubSuccess                            t2-ce-01.lnl.infn.it
5     N   Running           SubSuccess                            cream01.lcg.cscs.ch
```

```
6     N   Running           SubSuccess                          cream01.lcg.cscs.ch
7     N   Running           SubSuccess                          ingrid.cism.ucl.ac.be
8     N   Running           SubSuccess                          ingrid.cism.ucl.ac.be
9     N   Running           SubSuccess                          ce203.cern.ch

crab:   9 Total Jobs
 >>>>>>>>> 1 Jobs Submitted
        List of jobs Submitted: 2
 >>>>>>>>> 8 Jobs Running
        List of jobs Running: 1,3-9

crab:  You can also follow the status of this task on :
        CMS Dashboard: http://dashb-cms-job-task.cern.ch/taskmon.html#task=fanzago_crab_0_131014_
        Your task name is: fanzago_crab_0_131014_160518_582igd

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/log/crab.log
```

and then ...

```
$ crab -status
crab:  Version 2.9.1 running on Tue Oct 15 10:53:33 2013 CET (08:53:33 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TEST_RELEASE/TEST_PATC2/TEST_2_8

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Checking the status of all jobs: please wait
crab:  contacting remote host submit-4.t2.ucsd.edu
crab:  Establishing gsissh ControlPath. Wait 2 sec ...
crab:  Establishing gsissh ControlPath. Wait 2 sec ...
crab:
ID    END STATUS            ACTION       ExeExitCode JobExitCode E_HOST
----- --- ----------------- ------------  ---------- ----------- ---------
1     N   Done              Terminated    0          0           ce208.cern.ch
2     N   Done              Terminated    0          60317       cream03.lcg.cscs.ch
3     N   Done              Terminated    0          60317       cream03.lcg.cscs.ch
4     N   Done              Terminated    0          0           t2-ce-01.lnl.infn.it
5     N   Done              Terminated    0          60317       cream01.lcg.cscs.ch
6     N   Done              Terminated    0          60317       cream01.lcg.cscs.ch
7     N   Done              Terminated    0          0           ingrid.cism.ucl.ac.be
8     N   Done              Terminated    0          0           ingrid.cism.ucl.ac.be
9     N   Done              Terminated    0          0           ce203.cern.ch

crab:  ExitCodes Summary
 >>>>>>>>> 4 Jobs with Wrapper Exit Code : 60317
         List of jobs: 2-3,5-6
        See https://twiki.cern.ch/twiki/bin/view/CMS/JobExitCodes for Exit Code meaning

crab:  ExitCodes Summary
 >>>>>>>>> 5 Jobs with Wrapper Exit Code : 0
         List of jobs: 1,4,7-9
        See https://twiki.cern.ch/twiki/bin/view/CMS/JobExitCodes for Exit Code meaning

crab:   9 Total Jobs

crab:  You can also follow the status of this task on :
        CMS Dashboard: http://dashb-cms-job-task.cern.ch/taskmon.html#task=fanzago_crab_0_131014_
        Your task name is: fanzago_crab_0_131014_160518_582igd

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TEST_RELEASE/TEST_PATC2/TEST_2_8_2/TUTORIAL/TUT_
```

Job Output Retrieval

For the jobs which are in the "Done" status it is possible to retrieve the log files of the jobs (just the log files, because the output files are copied to the Storage Element associated to the T2 specified on the crab.cfg and infact return_data is 0). The following command retrieves the log files of all "Done" jobs of the last created CRAB project:

```
crab -getoutput
```

to get the output of a specific project:

```
crab -getoutput -c  <dir name>
```

the job results will be copied in the `res` subdirectory of your crab project:

```
$ crab -get
crab:  Version 2.9.1 running on Tue Oct 15 10:53:53 2013 CET (08:53:53 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/

crab:  error detecting glite version
crab:  error detecting glite version
crab:  contacting remote host submit-4.t2.ucsd.edu
crab:  Preparing to rsync 2 files
crab:  Results of Jobs # 1 are in /afs/cern.ch/user/f/fanzago/scratch0/TEST_RELEASE/TEST_PATC2/TE
crab:  contacting remote host submit-4.t2.ucsd.edu
crab:  Preparing to rsync 16 files
crab:  Results of Jobs # 2 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
crab:  Results of Jobs # 3 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
crab:  Results of Jobs # 4 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
crab:  Results of Jobs # 5 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
crab:  Results of Jobs # 6 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
crab:  Results of Jobs # 7 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
crab:  Results of Jobs # 8 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
crab:  Results of Jobs # 9 are in /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160
Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/log/crab.log
```

Use the -report option

As for the MonteCarlo data example, it is possible to run the report command:

```
crab -report -c <dir name>
```

the report command returns info about correctly finished jobs, that means jobs with JobExitCode = 0 *and* ExeExitCode = 0

```
$ crab -report
crab:  Version 2.9.1 running on Tue Oct 15 15:55:10 2013 CET (13:55:10 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/

crab:  error detecting glite version
```

```
crab:  error detecting glite version
crab:  --------------------
Dataset: /SingleMu/Run2012B-13Jul2012-v1/AOD
Remote output :
SE: T2_IT_Legnaro t2-srm-02.lnl.infn.it  srmPath: srm://t2-srm-02.lnl.infn.it:8443/srm/managerv2?
Total Events read: 264540
Total Files read: 21
Total Jobs : 9
Luminosity section summary file: /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_1605
   # Jobs: Retrieved:9

---------------------------

crab:  Summary file of input run and lumi to be analize with this task: /afs/cern.ch/user/f/fanza

crab:  to complete your analysis, you have to analyze the run and lumi reported in the //afs/cern

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TUTORIAL/crab_0_131014_160518/log/crab.log
```

where the content of files containing the luminosity info about the task are: the original lumiMask.json file written in the crab,.cfg file and used during the creation of your task

```
$ cat Cert_190456-208686_8TeV_PromptReco_Collisions12_JSON.txt
{"190645": [[10, 110]], "190704": [[1, 3]], "190705": [[1, 5], [7, 65], [81, 336], .... "208686":
```

the lumi sections that your created jobs have to analyze (that are info used as arguments of your jobs)

```
$ cat crab_0_131014_160518/res/inputLumiSummaryOfTask.json

{"194305": [[84, 85]], "194108": [[95, 96], [117, 120], [123, 126], [149, 152], [154, 157], [160,
```

the lumi sections really analyzed by your correctly terminated jobs

```
$ cat crab_0_131014_160518/res/lumiSummary.json
{"195947": [[27, 27], [36, 36]], "194108": [[95, 96], [119, 120], [123, 126], [154, 157], [160, 1
```

and the missing lumi (difference between the original lumiMask and lumiSummary) that you can analyze creating a new task and using this file as new lumiMask file

```
$ cat crab_0_131014_160518/res/missingLumiSummary.json file
{"190645": [[10, 110]],
 "190704": [[1, 3]],
 "190705": [[1, 5], [7, 65], [81, 336], [338, 350], [353, 383]],
 "190738": [[1, 130], [133, 226], [229, 355]],
.....
 "208541": [[1, 57], [59, 173], [175, 376], [378, 417]],
 "208551": [[119, 193], [195, 212], [215, 300], [303, 354], [356, 554], [557, 580]],
 "208686": [[73, 79], [82, 181], [183, 224], [227, 243], [246, 311], [313, 463]]}
```

To create a task to analyze the missing lumis of the original lumiMask you can use the missingLumiSummary.json file as new lumiMask.json file in your crab.cfg. As before, you can decide the split you want, and using the same publish_data_name the news outputs will be published in the same dataset of previuosly task

```
[CMSSW]
lumis_per_job           = 50
number_of_jobs          = 4
pset                    =  tutorial.py
datasetpath             = /SingleMu/Run2012B-13Jul2012-v1/AOD
lumi_mask             =  crab_0_131014_160518/res/missingLumiSummary.json
output_file            = outfile.root
```

```
[USER]
return_data              = 0
copy_data                = 1
publish_data =1
storage_element          = T2_xx_yyyy
publish_data_name        = FanzagoTutGrid_data

[CRAB]
scheduler               = remoteGlidein
jobtype                 = cmssw

$ crab -create -cfg crab_missing.cfg
[fanzago@lxplus0445 SLC6]$ crab -create -cfg crab_data.cfg
crab:  Version 2.9.1 running on Tue Oct 15 17:10:16 2013 CET (15:10:16 UTC)

crab. Working options:
        scheduler           remoteGlidein
        job type            CMSSW
        server              OFF
        working directory   /afs/cern.ch/user/f/fanzago/scratch0/TEST_RELEASE/TEST_PATC2/TEST_2_8

crab:  error detecting glite version
crab:  error detecting glite version
crab:  Contacting Data Discovery Services ...
crab:  Accessing DBS at: http://cmsdbsprod.cern.ch/cms_dbs_prod_global/servlet/DBSServlet
crab:  Requested (A)DS /SingleMu/Run2012B-13Jul2012-v1/AOD has 14 block(s).
crab:  SE black list applied to data location: ['srm-cms.cern.ch', 'srm-cms.gridpp.rl.ac.uk', 'T1
crab:  Requested number of jobs reached.
crab:  4 jobs created to run on 200 lumis
crab:  Checking remote location
crab:  WARNING: The stageout directory already exists. Be careful not to accidentally mix outputs
crab:  Creating 4 jobs, please wait...
crab:  Total of 4 jobs created.

Log file is /afs/cern.ch/user/f/fanzago/scratch0/TEST_RELEASE/TEST_PATC2/TEST_2_8_2/TUTORIAL/TUT_
```

and submit them as usual. The created jobs will analyze part of the missing lumi of the original lumiMask.json file.

- If you select total_number_of_lumis = -1 instead of lumi_per_job or number_of_job, the new task will analyze all the missing lumi.

Run Crab retrieving your output (without copying to a Storage Element)

You can also run your analysis code without interacting with a remote Storage Element, but retrieving the outputs to your workspace area (under the res dir of the project). Here below an example of the CRAB configuration file, coerent with this tutorial:

```
[CMSSW]
total_number_of_events  = 100
number_of_jobs          = 10
pset                    = tutorial.py
datasetpath             =  /RelValZMM/CMSSW_5_3_6-START53_V14-v2/GEN-SIM-RECO
output_file              = outfile.root

[USER]
return_data             = 1

[CRAB]
scheduler               = remoteGlidein
```

```
jobtype                 = cmssw
```

And with this crab.cfg in place you can re-do de workflow as described before (a part of the publication step):

- creation
- submission
- status progress monitoring
- output retrieval (in this step you'll be able to retrieve directly the real output produced by your pset file)

Where to find more on CRAB

- CRAB Home
- CRAB3 Tutorial
- HowTos
- CRAB FAQ
- WorkBookGridJobDiagnosisTemplate: Steps to identify the problems you experience with your grid analysis jobs.
- CRAB mailing list where to send feedback and ask support in case of jobs problem (please send to us your crab.cfg file and the job stderr - stdout - log otherwise we are not able to provide support)

Note also that all CMS members using the Grid must subscribe to the Grid Annoucements CMS.HyperNews forum.

Review status

Reviewer/Editor and Date (copy from screen)	Comments
JohnStupak - 4-June-2013	Review, minor revisions, updated real data dataset to an existing dataset
NitishDhingra - 2012-04-07	See detailed comments below.
MattiaCinquilli - 2010-04-15	Update for tutorial
FedericaFanzago - 18 Feb 2009	Update for tutorial
AndriusJuodagalvis - 2009-08-21	Added an instance of url_local_dbs

Detailed comments 07-Apr-2012 Hide
Complete Review, Minor Changes. Page gives a good idea of doing a physics analysis using CRAB

Responsible: FedericaFanzago

This topic: CMSPublic > WorkBookCRAB2Tutorial
Topic revision: r120 - 2016-04-19 - FedericaFanzago