Table of Contents

5.3 Data Analysis Work Flow

Complete:
Detailed Review status

Goals of this page

When you finish this page, you should understand:

- The steps that you need to follow in order to run an analysis job on grid resources.
- The basics about how CRAB (Cms Remote Analysis Builder) works.

This page does not teach you how to use CRAB. It only provides background material on how things work.

To learn how to use CRAB see Chapter "Analysis with CRAB".

Contents

Introduction

Data Analysis in CMS involves the following steps:

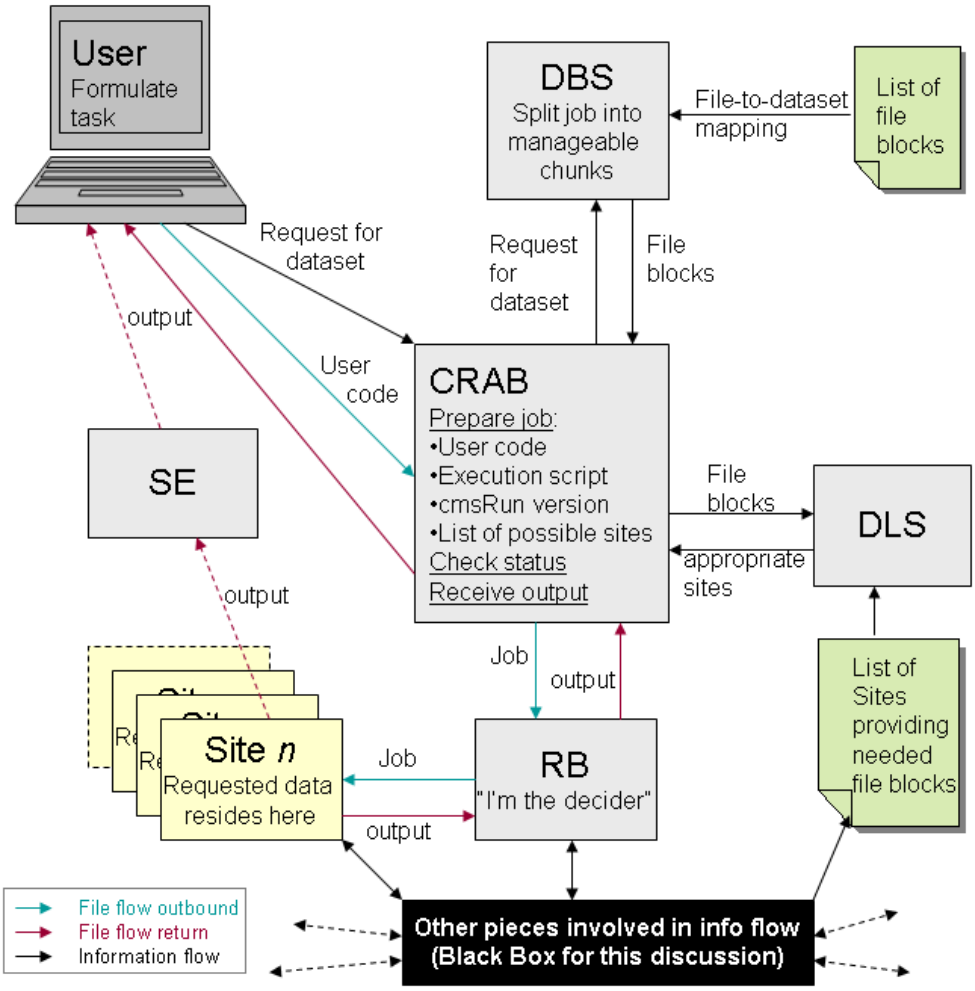
- Developing an executable to run on data.
- Testing that executable locally on your desktop/laptop/lxplus by running it on at least one file from the dataset you want to run on.
    - See Chapter "Locating Data" for details of how to find data, and pull a few files to your desktop/laptop.
- Doing the actual data analysis with CRAB
    - See Chapter "Analysis with CRAB" for details.

CRAB is a Python program intended to simplify the process of creation and submission of CMS analysis jobs into a grid environment. You'll use it to run your jobs on the grid (LCG or OSG). The remainder of this page will explain what CRAB does under the hood. This is background information for you to better understand what you are doing when you use CRAB.

Workflow Illustration

The figure below shows the flow of user code, physics data, and job- and resource-related information throughout the course of an analysis job. While this figure was drawn in 2006, it is still correct, Analysis Workflow has not changed since the start of CMS. You may want only to read:

- DLS (DataLocationService) as: PhEDEx
- RB (ResourceBroker) as: Grid Scheduler meaning something that submits jobs to Grid resources, as of 2016 we use HTCondor global pool



Task Formulation by the user

The first steps are required for any analysis:

```
# make a your working directory
   mkdir MYDEMOANALYZER
   cd MYDEMOANALYZER



# if output of echo $0 is csh or tcsh
   setenv SCRAM_ARCH slc7_amd64_gcc820

# if output of echo $0 is bash/sh
```

```
   export SCRAM_ARCH=slc7_amd64_gcc820

# check your arch - it should give an output as slc7_amd64_gcc820
   echo $SCRAM_ARCH


 # create a new project area
 cmsrel CMSSW_10_2_18
 echo $0


 cd CMSSW_10_2_18/src/
 cmsenv
```

Write a Framework Module

First, create a subsystem area. The actual name used for the directory is not important, we'll use `Demo`. From the `src` directory, make and change to the `Demo` area:

```
mkdir Demo
cd Demo
```

Note that if you do not create the subsystem area and create you module directly under the `src` directory, your code will not compile. Create the "skeleton" of an EDAnalyzer module (see SWGuideSkeletonCodeGenerator for more information):

```
mkedanlzr DemoAnalyzer
```

Further steps for running parallel jobs on the Grid or on any batch processing system are:

1. Determine how to split your job into "chunks" that can run in parallel and finish in a reasonable amount of time (e.g., a few hours).
2. Create your CRAB configuration file, `crab.cfg`. In it, you tell CRAB where to get the code and the data, and how to split the job.
3. Submit the job to the Grid via CRAB.
4. Monitor your job, as needed.
5. Collect your output, create your plots, and make discoveries!

Job Preparation by CRAB

Data Discovery

CRAB performs a query to the Dataset Bookeeping System (DBS) to find the right data to access. To select the data, the user can to go to the DAS search page and select the data he/she is interested in by using the query functionalities. The result of this query is a list of `datasetpath`, in the form `/PrimaryDataset/ProcessedDataset/DataTier/` such as `/DY1JetsToLL_M-10To50_TuneZ2Star_8TeV-madgraph/Summer12-PU_S7_START52_V9-v1/AODSIM`. This datasetpath should be written in the crab configuration file `crabConfig.py`. On task creation, (a task is the collection of identical jobs which are created and eventually submitted to analyze a give set of data; the only difference among the jobs in a task is the events each job processes, as determined by the splitting) CRAB queries DBS for the datasetpath, and gets back the details of the dataset, such as number of events, number of files, number of events per file, etc. The result of the query is a list of event collections, grouped by the underlying file blocks to which the data correspond. Note that at this stage the tool doesn't need to know about

the exact data location or about the physical structure of the event collections; this will only be needed further down in the workflow. Note that the user does not need to know at all the location(s) of the data, this is dealt with internally by CRAB.

Job splitting

At this stage, CRAB can decide how (and if) to split the complete set of event collections among several jobs, each of which will access a subset of the event collections in the selected dataset, according to user requirements. The splitting mechanism will take care to configure each job with the proper subset of data blocks and event collections. The user's `crab.cfg` file must specify the criteria by which the job splitting will take place (e.g., maximum number of events per job, maximum number of jobs, etc.). The actual splitting might not follow precisely the user requirement due to physical data placements on files: in any case the total number of events will be that required by the user.

Job configuration

The Workload Management System (WMS) will create job configurations for every job which is to be submitted. There are in fact two levels of job configuration: the first for the CMS software framework, the second for the Grid WMS. The Grid one is entirely dealt with by CRAB, while the CMS software one is the one setup by the user and CRAB just modifies it in order to access data on the Grid.

Job submission

After the previous step, two configuration files exist for every job in the task:

- one for the application framework, and
- one for the Grid WMS.

At submission time, the submission tool will have information about data location, and will pass this information to the Grid Workload Management System (as of 2014 we only use HTCondor via the glideInWMS) which in turn can decide where to submit, according to some resource availability metrics. The CMS WM tools will submit the jobs to the Grid WM System, as a "job cluster" if necessary, for performance or control reasons, and will interact with the job bookkeeping system to allow the tracking of the submitted job(s). The submission can be direct (for a small task) or via a CRAB server, a CMS specific layer between user and the grid. In the latter case, the CRAB client, the one used by the user in the user interface, will pass the task specs to a CRAB server, which in turn will take care of submission to grid WMS (or local scheduler) on behalf of the user. The server will manage the task, monitor the jobs and eventually retrieve the output. The user will interact with the server rather than directly with the grid.

Job scheduling

The Grid WM System is responsible for scheduling the jobs to run on specific Computing Elements (CE) and dispatching them to the CE.

Job run-time

Job run-time takes place on a Worker Node (WN) of a specific Computing Element (CE). The jobs arrive on the WN with an application configuration which is still site-independent. The CE/WN is expected to be

configured such that the job can determine the locations of necessary site-local services (local file replica catalogue, CMS software installation on the CE, access to CMS conditions, etc.).

Job completion

Once the job completes, it must store its output someplace. For very small outputs, the outputs may just be returned to the submitter as part of the output sandbox. For larger outputs, the output can be stored on the local Storage Element (SE) (for subsequent retrieval by the user): given a limitation in size of the output sandbox, any output larger than a few MB have to be copied to a remote SE and not returned via sandbox. The job's only obligation is to either successfully store the outputs to the local SE or pass them to the data transfer agent. It is assumed that the Grid WM System will handle the task of making the output sandbox, log files, etc., available to the user.

Task monitoring

While processing is in progress, the user can monitor the progress of the jobs constituting his or her task by using the job bookkeeping and monitoring system (`crab status`). Additional information about task status (also historical), can be found on Dashboard

Task completion

As individual jobs finish (or after the entire set of jobs in the task has finished) the user will find the resulting output data coalesced to the destination specified during the "job completion" step, above. A list of the runs and luminosity sections read in input is also available to determine the luminosity this analysis corresponds to. If the user wishes to publish this data, the relevant provenance information must be extracted from the job bookkeeping system, etc., and published in DBS.

These pieces thus constitute a basic workflow using the CMS and Grid systems and services. The CMS WM tools are responsible for orchestrating the interactions with all necessary systems and services to accomplish each specified task.

Information Sources

- CMS Remote Analysis Builder - CRAB
- CPT Technical Design Report Section 4.8.1.

Review status

Reviewer/Editor and Date (copy from screen)	Comments
StefanoBelforte - 2017-07-04	Slightly update to make it valid in CRAB3 + HTCondor world
JohnStupak - 4-June-2013	Minor revisions and update to 5_3_5
NitishDhingra - 29-Mar-2012	See detailed comments below
StefanoBelforte - 11-Nov-2010	Add information on luminosity of results

StefanoBelforte - 22-Jan-2010	Complete Expert Review, no changes
FrankWuerthwein - 06-Dec-2009	Complete Reorganization 1st draft ready for review
SimonMetson - 28 Feb 2008	review: Updated DBS Discovery link. In the (near) future this page should be updated to refer to the CRAB server
StefanoLacaprara - 1 Feb 2008	review: fill uptodate information plus add link to DBS and dashboard
StefanoLacaprara - 16 Nov 2006	review: minor mods and add comments about what is not yet possible with CRAB
AnneHeavey - 23 Jun 2006	Significant editing; move this from Intro down to Using the Grid

Detailed comments 29-Mar-2012 Hide
Complete review. Added information on deprication of DBS, added link to DAS, fixed broken links. The information on the page is quite clear.

Responsible: DaveEvans
Last reviewed by: SimonMetson - 28 Feb 2008

This topic: CMSPublic > WorkBookAnalysisWorkFlow
Topic revision: r39 - 2020-11-14 - NitishDhingra