



CRAB tutorial

19/02/2009

Marco Calloni CERN – Milano Bicocca

Federica Fanzago INFN Padova



Outline



- short CRAB tool presentation
- hand-on session
- trouble shooting



Prerequisites



- We expect you know:
 - Howto run CMSSW codes locally
 - Howto discover where data you want to analyze are
 - How to get a user proxy
 - You are registered to the CMS VO
- If at least one of the previous points is not true, please look at the CMS workbook
 - <https://twiki.cern.ch/twiki/bin/view/CMS/WorkBook>



CRAB for newbies



- What is the CMS Remote Analysis Builder?
 - Is a user-friendly command line tool that let you interact with the Grids & the Local Resources (LSF/CAF)
 - Takes care of hiding you most of the distributed model complexities
 - Lets you run CMSSW over whole datasets already distributed among the sites
 - Automate most of the analysis workflow (status tracking, resubmissions, ...)



How CRAB works



- CRAB does not compile on the remote sites
 - What you have locally is what you get worldwide
- CRAB takes care of reproducing your whole working environment remotely
- The only relevant information CRAB needs
 - which version of CMSSW you refer to
 - the ParameterSet
 - the Dataset name you'r going to analyse



What CRAB does



- Create tasks preparing the jobs for the remote environment and split the load according your requests
- Submit the created jobs
- Monitor the progress of the jobs
- Handle user output moving them to your reference SE
- Resubmit failed jobs
- Notify about task progress via e-mail
- Publish the result to DBS



How you can talk to CRAB



- Every configuration in CRAB is set through the directives reported in the crab.cfg file
 - Organized as key = value pairs
 - Grouped in macro-sections [CRAB], [CMSSW], [USER], ...
- A minimal and a full template for crab.cfg are in
\$CRABPATH/crab.cfg,
\$CRABPATH/full_crab.cfg
- Inline documentation (crab -h) guides you to set attributes



Some useful commands



- Create the CRAB project (by default crab.cfg)
 - `crab -create`
- Submit your jobs
 - `crab -submit <all | n | rng > [-c <crab_prj>]`
- Track the jobs progress
 - `crab -status [-c <crab_prj>]`
 - alternative use of the CRABSERVER web interface
- When jobs get done, retrieve data:
 - `crab -getoutput <all | rng > [-c <crab_prj>]`
 - output will store in `<crab_prj/res>`



Some useful commands



- If you need to kill some job
 - `crab -kill <all | n | rng > [-c <crab_prj>]`
- Get post-mortem infos (useful for abort jobs)
 - `crab -postMortem <all | rng > [-c <crab_prj>]`
- Resubmit
 - `crab -resubmit <all | rng > [-c <crab_prj>]`
- Publish your results, if you need to share them
 - `crab -publish [-c <crab_prj>]`
- Clean the obsolete CRAB project
 - `crab -clean [-c <crab_prj>]`



Publication



- Essential step to re-run with Grid and CRAB
- To take care before to publish your data
 - You must be registered in SiteDB
 - You must know the local DBS instance where to publish
 - You must know a Tier2 StorageElement where to store data
- Moreover
 - crab.cfg must contain the publication directives BEFORE creation The .root must be an EDM file
- <https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrabForPublication>



CRAB documentation



- The inline manual
 - `crab -h` for crab configuration syntax
- HowTo and FAQ Twiki pages
 - <https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrabHowTo>
 - <https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrabFaq>
- The CRAB twiki page
 - <https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrab>
- Get support from `hn-cms-crabFeedback` mailing-list (please add your `stderr-stdout-log` and `crab.cfg`)