

CMS data distributed analysis with CRAB3



Marco Mascheroni

INFN Milano Bicocca

On behalf of the CRAB3 Team

CHEP, Okinawa, April 2015



- 1 What and Why
- 2 Architecture Overview
- 3 Commissioning and Adoption



CRAB3: Motivations

User Analysis in CMS

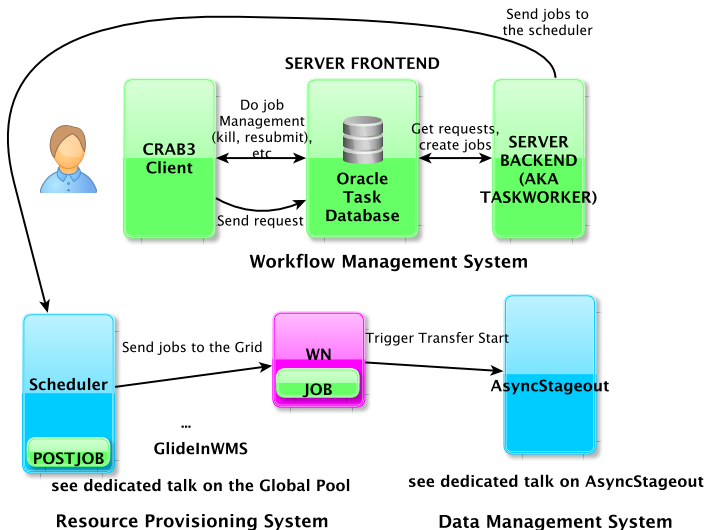
- "User Analysis" is an important activity in every high energy physics experiment and CMS is no exception
 - Goes from generating personal simulated events to transforming CMS-produced events into private formats
- Version 2 of the CMS Remote Analysis Builder (CRAB2) was the official tool used for accessing Grid resources during Run1.

A new Tool

- Profiting from the lower activity during LS1, CRAB2 was completely rewritten to overcome the limitations found during Run1:
 - Chaotic transfer architecture
 - Heavy client (both time and code wise)
 - Poor user experience (e.g.: no automatic job resubmission)



Architecture Overview





The CRAB3 client

What it is:

- Command line interface used by physicists to access the infrastructure
- Lightweight, modular and pluggable
 - Only *python* and *pycurl* are required to run it
 - Libraries to facilitate scripting for users are provided
 - Easy to add new commands
- Distributed through *cvmfs* with other CMS software

From a functional perspective:

- Creates user sandbox, few splitting parameters and gives everything to the server
- Allow users to check status of the jobs (which are grouped in *tasks*)
- Allow users to do task management (kill/resubmit/output retrieval)



The CRAB3 Server

The Frontend

- Act as a gateway for user requests
 - Handles authentication through https (X509 certificates)
- Validate user requests, cache them in the Oracle DB
- Exposes a REpresentational State Transfer interface
 - Clear separation of data and interface (easy to switch DBMS)
- Deployed on the CMSWEB cluster and operated by the HTTP group in CMS

The Backend (aka TaskWorker)

- Essentially a big queue where user requests are queued and processed
- Does data discovery, and job splitting, creation, and submission



Other

Postjob

- Runs on the scheduler machine
- Waits for the transfer to finish in ASO, resubmit the job if necessary

Async StageOut (ASO)

- Moves the output files from the site where the job ran, to the destination
- It uses FTS (File Transfer System), a low-level data movement service developed by CERN-IT

GlideInWMS Global Pool

- Resource (pilot) management system of CMS. Takes care of providing slots depending on the requested load
- Shared with the central production team



Commissioning and adoption

Opportunity to do an extensive test of CRAB3 during the Computing, Software, and Analysis Challenge (CSA14)



The CSA14 Challenge

What is CSA14

CSA14 was a series of large-scale tests of the complete CMS data processing, software, and analysis chain organized from June to September 2014. The whole CRAB3 system (including ASO and global pool) was part of the tests.

In particular, a scale and a functional tests were organized for CRAB3:

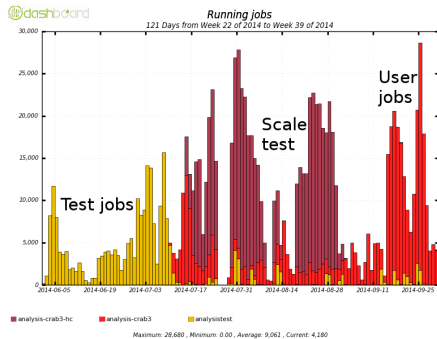
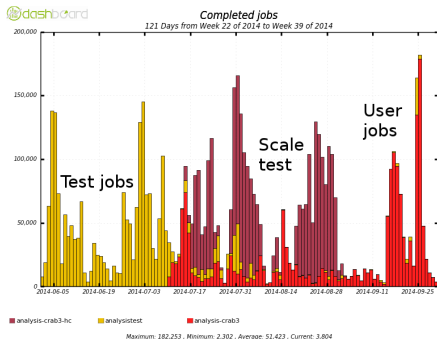
Scale test goal

Check how the system reacts under high load: target of 20k parallel running jobs and 200k jobs per day.

Functional test goal

Expose the system to the users through a miniAOD production campaign (new CMS data format): useful to collect feedback and test new use cases.

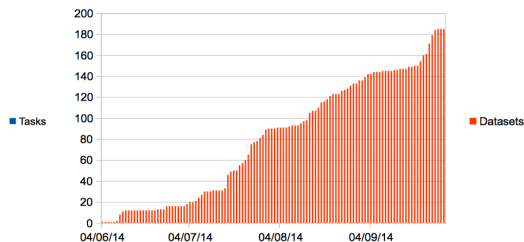
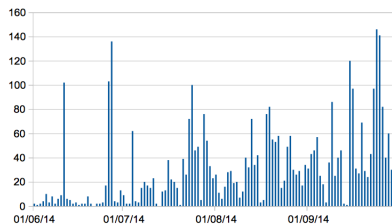
Scale tests results



- User activity integrated with test jobs (using HammerCloud)
- Reached 30k parallel running jobs and almost 200k jobs per day!
 - And there is still room for improvements, the system was not pushed to its limit to not interfere with real user analysis.



Functional Test Results



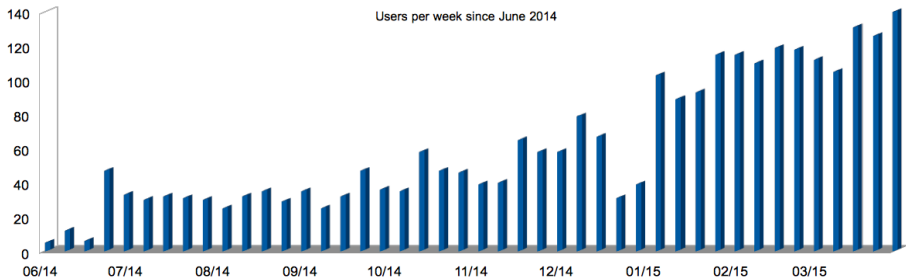
- First time the system was exposed to users (200 users tried CRAB3 during that time)
- More than 180 datasets were published in the CMS bookkeeping system (DBS) in the context of CSA14 miniAOD productions

Operational/infrastructural issues encountered and promptly solved, useful feedback collected (and implemented) from users.



Adoption plan

- CRAB3 is the default system since 2015 (last CRAB2 release was on Nov the 20th).
 - Support is limited to the cases not covered by CRAB3 yet (i.e.: access to local resources)
- User migration from CRAB2 is continuing well and the number of people using the tool is steadily growing.





Support, Operations, and Further Developments

Communications with Users

- Communications goes through the hypernews forum dedicated to the CMS computing tools
- Tutorials for new users are regularly organized
- Feedback is collected from the users and a monthly release schedule has been established for deploying new developed features.

Operations

- Many types of monitoring/alarms are in place or they are being developed
 - Monitor systems for users (glidemon, dashboard)
 - Monitoring machines performance through ganglia (#jobs, memory/cpu usage, I/O load etc)
 - Monitor for debugging user problems (on top of the REST interface)



Conclusions and Future Works

Conclusions

- Development and commissioning of CRAB3 has been a success
- Many improvements thanks to the new architecture:
 - Better user experience (e.g.: automatic resubmission)
 - Transfers centrally managed (major failure reason in CRAB2)
 - Improved operability/sustainability (more code deployed centrally)

Future Work

- Improve the management of user-data
 - A review with the users has been held and set recommendations for actions to be carried on before or during Run2
- Continue with the established monthly release process to steadily improve the tool
 - Both add new features and improve stability/operability