CRAB: the CMS distributed analysis tool on EGEE/WLCG infrastructure

Federica Fanzago for the CRAB team

INFN Padova

Outline

- **CMS introduction**
  - CMS computing model
- **The distributed analysis**
  - Requirements
- **CRAB**
  - The aim
  - How it works
- **CRAB usage**
  - Test and data challenge
- **CRAB improvements**
- **Conclusion**

Introduction

- CMS “Compact Muon Solenoid” is one of the four particle physics experiment that will collect data at LHC “Large Hadron Collider” at CERN
- The large amount of produced data (events) needs to be accessed for analysis from world-wide distributed physicists

“bunch crossing” every 25 nsecs.
100 “triggers” per second
Each triggered event ~1 MB in size

- CMS will produce
  - ~2 PB events/year (assumes startup luminosity 2x1033 cm-2 s-1)
- All events will be stored into files
  - O(10^6) files/year
- Files will be grouped in Fileblocks
  - O(10^3) Fileblocks/year
- Fileblocks will be grouped in Datasets
  - O(10^3) Datasets (total after 10 years of CMS)
  - 0.1- 100 TB

Issues and help

- **How to**
  - store and manage this huge quantity of data ?
  - assure data access to physicists of CMS collaboration ?
  - assure enough computing power for data analysis and simulation ?
  - guarantee resources and data availability to allowed users ?

  CMS uses a distributed architecture based on Grid infrastructure

- **Tools for accessing distributed data and resources are provided by WLCG (World LHC Computing Grid) with two main different flavours**
  - LCG/gLite in Europe, OSG in the US

The CMS computing model

The CMS offline computing system is arranged in hierarchical Tiers geographically distributed.

Online system

Offline farm

recorded data

Data from DAQ are sent, stored and first step reconstructed at Tier-0, then spread over T1s

Tier 0

CERN Computer center

Tier 1

France Regional Center

Italy Regional Center

Fermilab Regional Center

T1s take care about
- calibration, skimming and reconstruction. They sent data to T2s

Tier 2

Tier2 Center

Tier2 Center

Tier2 Center

. . .

T2s provide power for analysis and simulation

Tier 3

InstituteA

InstituteB

workstation

Remote data accessible via grid

Distributed analysis requirements

- **The analysis in a distributed environment is a complex computing task because it assume to know:**
  - which data are available
  - where data are stored and how to access them
  - which resources are available and are able to comply with analysis requirements
  - Grid and CMS infrastructure details

- **Users point of view:**
  - Want to analyze distributed data as they were in the local farm
  - Don’t want to became Grid experts

How to allow the distributed analysis

- **The CMS collaboration is developing some tools interfaced with Grid services to simplify the analysis task, including**
  - Agents for automatic data distribution among tiers
  - Catalogs for data location (Data Bookkeeping System)
  - CMS software distribution and installation

- **But this is not enough ….how to help users to run their analysis code on data available only at remote sites?**
  - CRAB, the CMS Remote Analysis Builder.

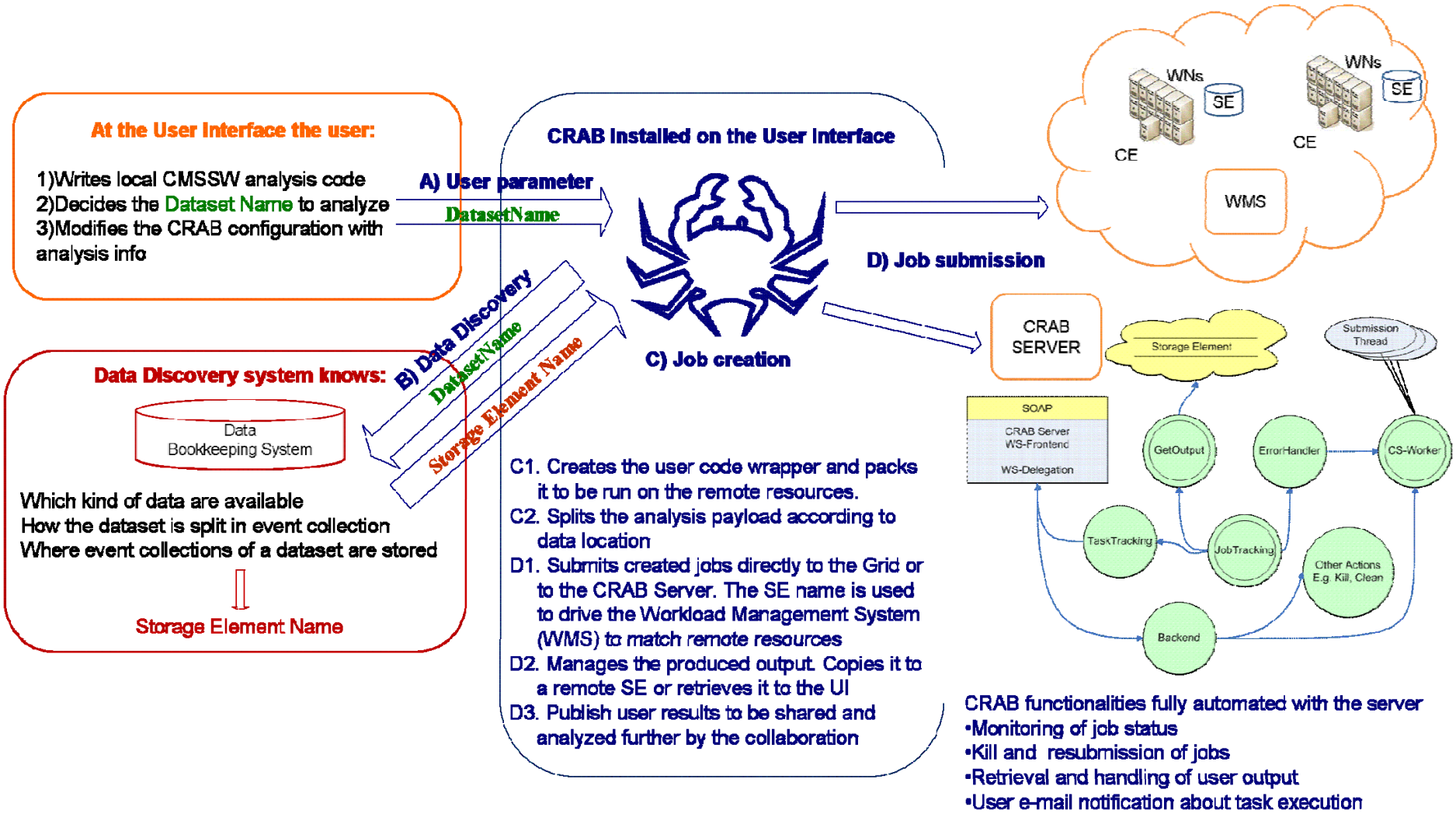
CRAB

- **The aim of CRAB is to simplify the work of users to create, submit and manage their analysis job in Grid environments, hiding Grid and CMS infrastructure details**
  - Users have to develop their analysis code in a interactive environment, to test it on few local data and then decide which remote data to analyse.
  - CRAB replicates user analysis environment into the remote resources.
  - CRAB handles data discovery, resources availability, job creation and submission, status monitoring and output retrieval. It allows also the publication of users analysis results to the local DBS catalogues.

The user-friendly tool

- **CRAB is a user-friendly tool**
  - Simple to install (distributed as tar archive)
  - Users have to modify the CRAB configuration file to provide analysis information (dataset name, n.of events, n.of jobs, how to manage produced output…)
  - Command line for actions (under user control)
    - >>crab -create
    - >>crab -submit (…. -status, -getOutput, -kill….)
  - Requirements
    - To interact with the Grid services and resources: CRAB must be installed in a UI, and user must have a valid Grid proxy
    - To interact with CMS environment: the CMS software has to be available in the UI

CRAB flow


At the User Interface the user:
1)Writes local CMSSW analysis code
2)Decides the Dataset Name to analyze
3)Modifies the CRAB configuration with analysis info
A) User parameter
DatasetName
CRAB Installed on the User Interface
C) Job creation
D) Job submission
WNs
SE
CE
WNs
SE
CE
WMS
Data Discovery system knows:
Data Bookkeeping System
Which kind of data are available
How the dataset is split in event collection
Where event collections of a dataset are stored
Storage Element Name
B) Data Discovery
DatasetName
Storage Element Name
C1. Creates the user code wrapper and packs it to be run on the remote resources.
C2. Splits the analysis payload according to data location
D1. Submits created jobs directly to the Grid or to the CRAB Server. The SE name is used to drive the Workload Management System (WMS) to match remote resources
D2. Manages the produced output. Copies it to a remote SE or retrieves it to the UI
D3. Publish user results to be shared and analyzed further by the collaboration
CRAB SERVER
Storage Element
Submission Thread
SOAP
CRAB Server WS-Frontend
WS-Delegation
GetOutput
ErrorHandler
CS-Worker
TaskTracking
JobTracking
Other Actions E.g. Kill, Clean
Backend
CRAB functionalities fully automated with the server
•Monitoring of job status
•Kill and resubmission of jobs
•Retrieval and handling of user output
•User e-mail notification about task execution

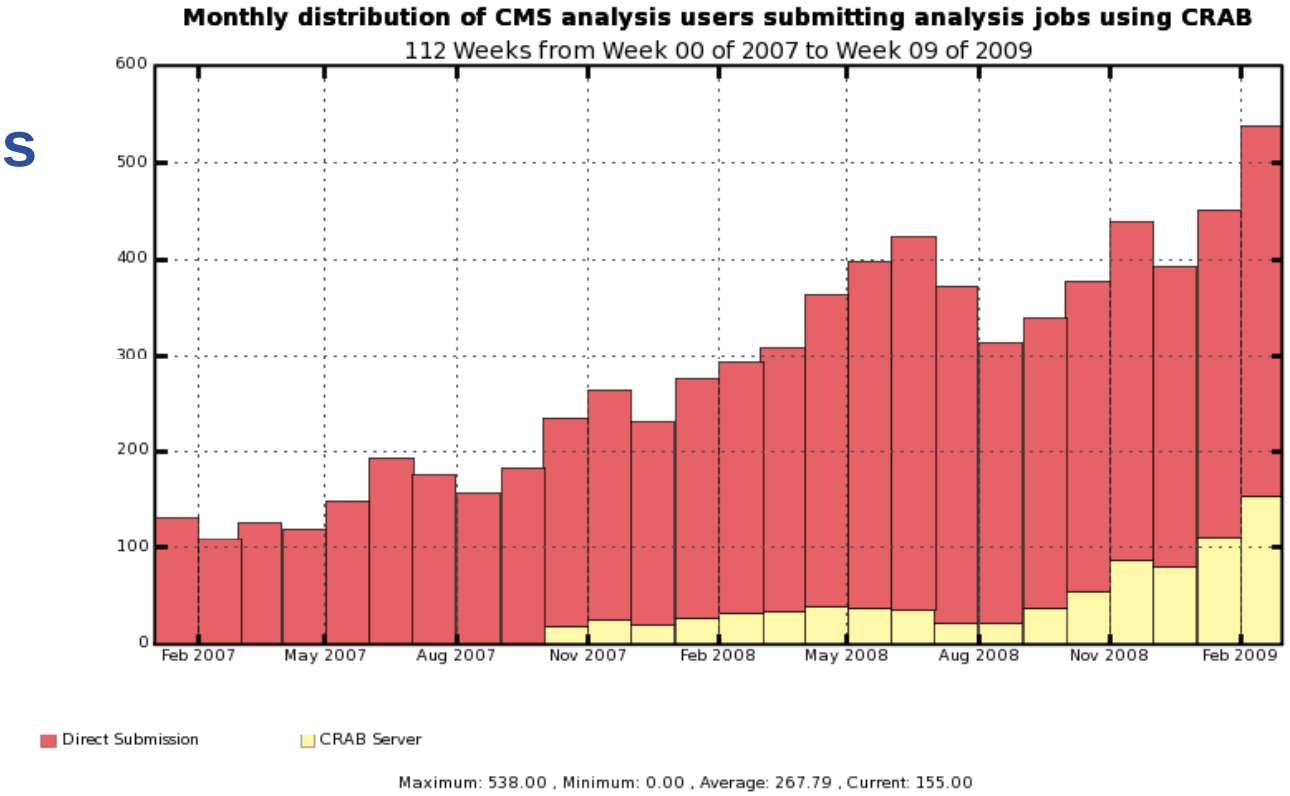
General CRAB info

- **CRAB was born in April 2005**
- **It is written in python**
- **Users appreciated the tool, providing feedback and asking for improvements**
- **CRAB client-server was released at end of 2007**
- **CRAB was used by many CMS collaborators to analyze remote data for CMS Physics TDR, otherwise not accessible**
- **Involved in most of the CMS Challenges**
  - the Magnet Test Cosmic Challenge
  - the Computing, Software, and Analysis Challenges over millions of simulated events, CSA06, CSA07, CCRC/CSA08

CRAB community

- **CRAB users community**
  - ~600 users
  - The average daily number ~70

- **Their number is increasing**
- **The submitted jobs too…**



Why a CRAB server?

- **To increase the scalability of the whole system to comply with the CMS requirements (100 kjobs/day)**
- **To reduce the user load automating most of the action done via command line, as submission, error handling, output retrieval, resubmission**
- **To improve the reliability of the system delegating to the server jobs handling**

- **Constraints:**
  - direct and Server submission modes must be transparent
  - same configuration file, same commands interface
  - simple switch from standalone to server mode and vice versa

CRAB server architecture

- **The server adopts a modular software approach:**
  - independent components implemented as agents
  - where needed, multithreading approach in the agent implementation
  - communication through an asynchronous and persistent message service (publish & subscribe model)

- **The system core is a MySQL DB (message service, job logging and bookkeeping …)**
  - other external components used for data storage and transfer:
    - default GridFTP but also: rfio, dcache…

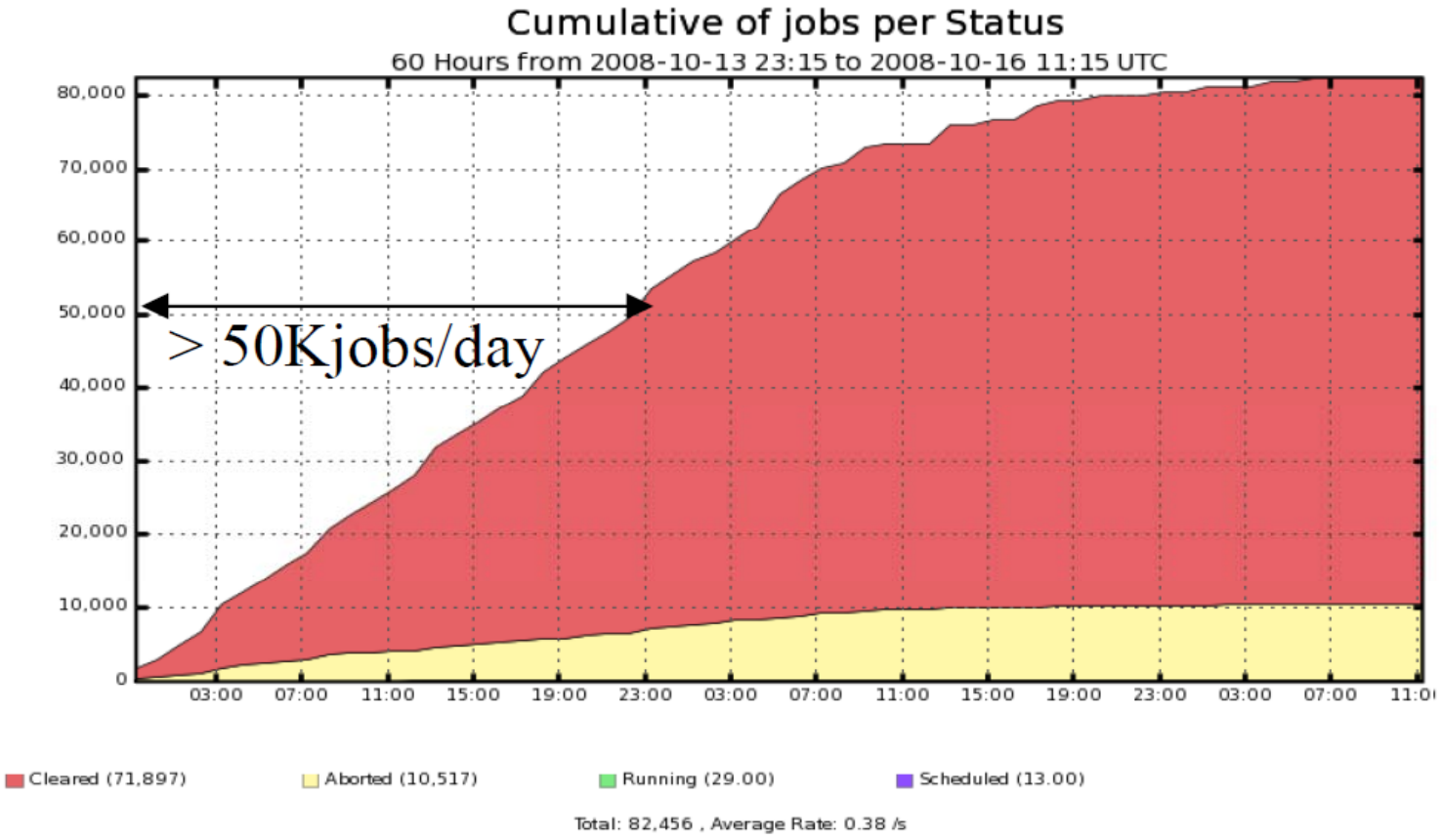
- **The architecture is as similar as possible to the other CMS Workload Management Tools: reusing code, sharing efforts**

CRAB server test (single user)

- **Very short jobs not reading an input dataset**
- **Zipped Sandbox of about 8MB**
- **Submitted to all sites excluding T1s**
  - Constant rate : 500-600 jobs every 20min
  - Plus Peaks of 1000-2000 jobs every 5 hours
- **Submitted from 40Kjobs/day to 50Kjobs/day**

CRAB server test (single user results)

- ~ 200Kjobs handled
- Job submitted to more than 50 CEs
- CRAB Server scales with 2 WMS!



CRAB server test (multi user)

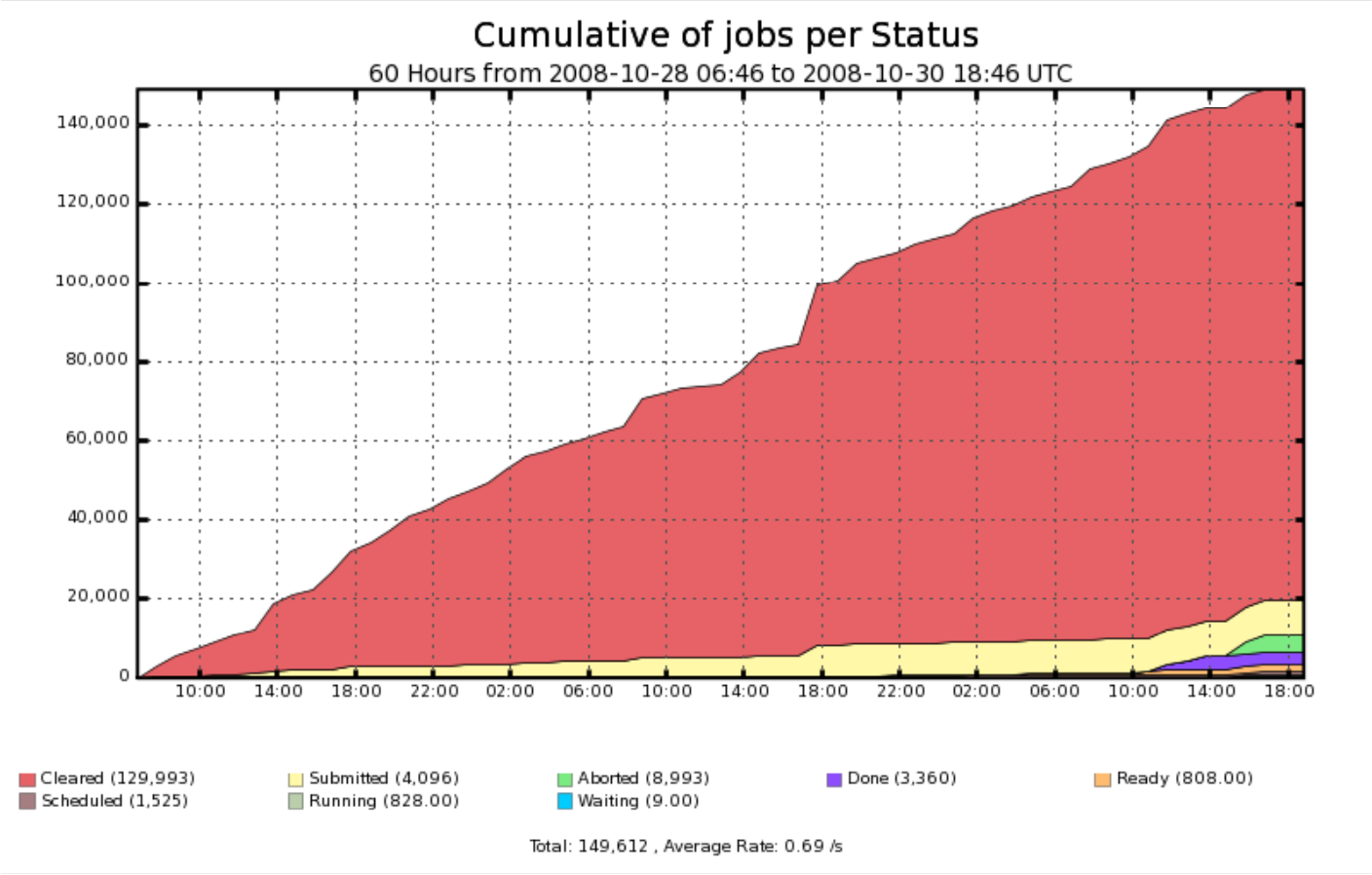
- **Job submission from different user certificates using more WMSs**

- **6 users with a variety of submission patterns:**
  - multi users scheduled submission:
    - 1 user submitting a task of 100jobs every 15min
    - 1 user submitting a task of 500 jobs every 20min
    - 1 user submitting a task of 600 jobs every 3hours
    - 1 user submitting a task of 2000 jobs every 6hours
  - random submissions: 2 volunteer users submitting at their will

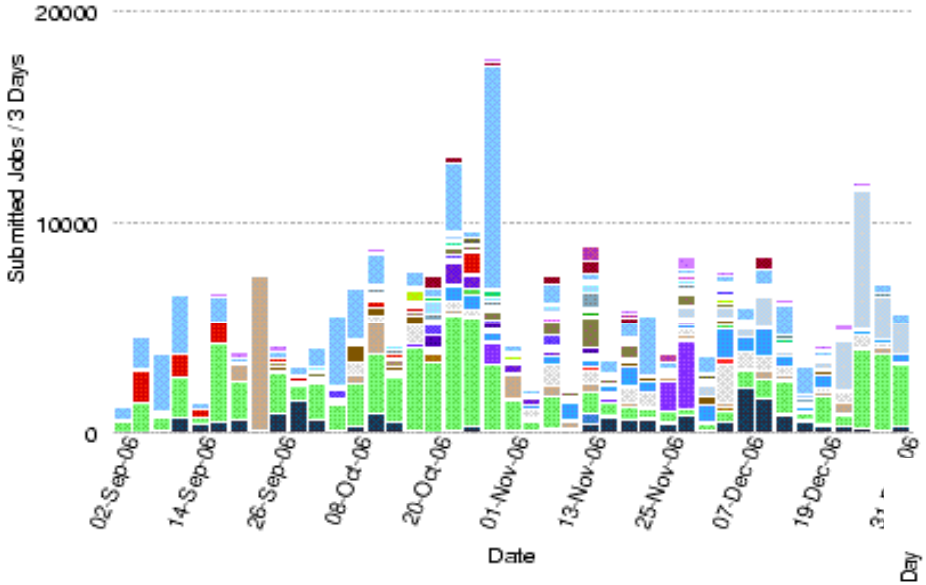
- **Increase in the DB content, starting from a not empty DB (about 200Kjobs from previous testing)**

CRAB server test (multi user results)

- **120Kjobs in 48 hours (with 2 WMS) with peaks of 2-3 Hz**
- **No evident problems with DB and gridFTP server**

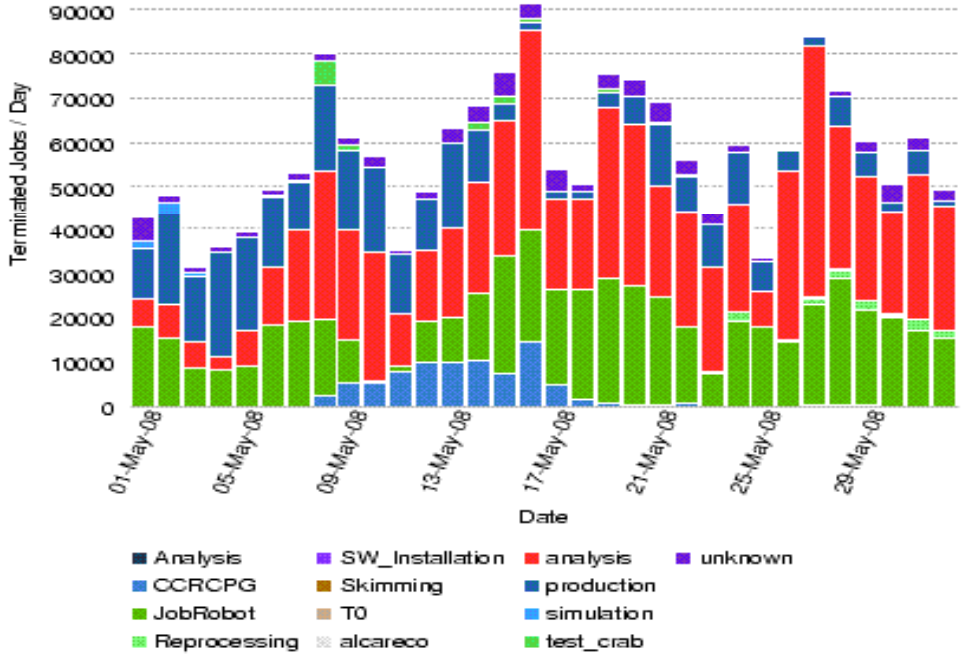


CRAB usage during challenge



- **CSA 2006 (last 2006 quarter):**
  - jobs/day peak: 18 kjobs
  - Average rate: 9 kjobs/day

- **CCRC may 2008:**
  - WLCG-wide readiness challenge
  - 1st challenge adopting CRAB Analysis Server for end-users processing



Reason of job failures

- **Grid infrastructure**
  - congestion of WMS
  - problem with the copy of output file to Storage Element (20%)
  - wrong site installation
- **CMS infrastructure**
  - CMSSW software not available
  - CMSSW misleading error code
  - problems with published dataset (some input file not found)
- **User code not correct**

CRAB improvements

- **Error handler plugin: the implementation allows the automatic resubmission of failed jobs depending on error reason.**
  - Improvement of actions for resubmission
  - Add info about sites and services status from distinct monitoring systems
- **Automatize user data publication**
- **Improve user interface (web)**
- **Add special analysis data flow for particular case, as data stored “runtime” in a SE**

Conclusion

- CRAB was born in April ’05
- A big effort to understand user needs and how to use in the best way services provided by Grid
- Lot of work to make it robust, flexible and reliable → from CRAB “Standalone” to CRAB Client-Server
- Users appreciate the tool and are asking for further improvements
- The use of CRAB proves the distributed analysis works for a generic CMS user !

Documentation

- **CRAB homepage**
  - https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrab
- **CRAB faq and howto**
  - https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideCrab#FAQ_HOWTO_Diagnosis_template
- **CRAB mailing list for users**
  - https://hypernews.cern.ch/HyperNews/CMS/get/crabFeedback.html