CRAB: the CMS tool to allow data analysis in a distributed environment

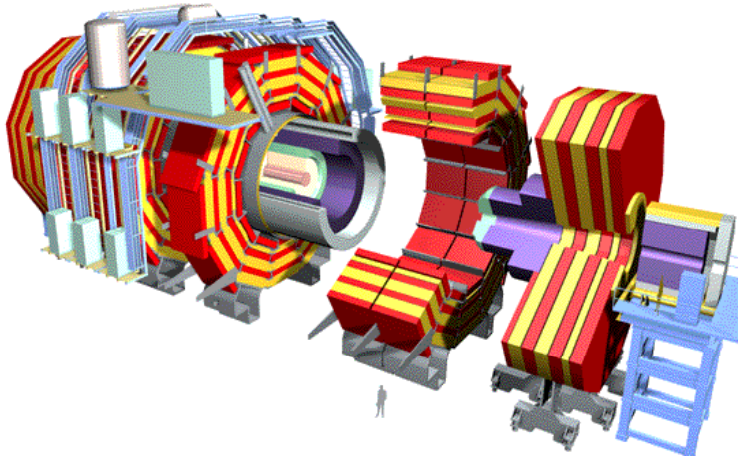
Federica Fanzago for the CRAB team
CERN - CNAF

Outline

- **CMS introduction**
  - CMS computing model
- **Distributed analysis**
- **CRAB "standalone"**
  - the flow
  - CRAB usage
- **The evolution of CRAB → the client – server architecture**
  - the flow
  - the implementation
  - CRAB client – server usage
- **Conclusion**

Introduction

- **CMS "Compact Muon Solenoid" is one of the four particle physics experiment that will collect data at LHC "Large Hadron Collider" starting in 2008 at CERN**
- **The large amount of produced data (events) should be available for analysis to 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

- **In order 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

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

- **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

Analysis chain

- **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
  - CMS software distribution and installation
  - CRAB user-friendly tool to help users to run their analysis code on data available at remote sites, hiding Grid and CMS infrastructure details

CRAB

- The aim of CRAB (Cms Remote Analysis Builder) is to simplify the work of users to create, submit and manage their analysis job in Grid environments.
- Users have to develop their analysis code in a interactive environment and decide which data to analyse.
- CRAB handles data discovery, resources availability, job creation and submission, status monitoring and output retrieval.

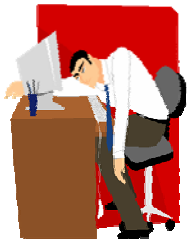
The user-friendly tool

- **CRAB is a user-friendly tool**
  - Simple to install (distributed as tar archive)
  - Only a configuration file where to provide analysis information
  - 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
    - To interact with CMS environment: the CMS software has to be available in the UI

CRAB "standalone" flow

The user working in the User Interface:

1)writes his analysis code as in a local environment
2)decides which data to analyze (Dataset Name)
3)modifies the CRAB configuration file with analysis info



C++ user code

CRAB.cfg:

Dataset Name
Number of events
Number of jobs
How to manage produced output

Data Discovery system:

Data Bookkeeping System

Data Location System

Which kind of data are available in the distributed environment and how the dataset is split in event collection

Where event collections of a dataset are stored in the distributed environment

Storage Element Name

CRAB is installed on the User Interface

1 User parameter

2 Data Discovery

Dataset Name

Storage Element Name

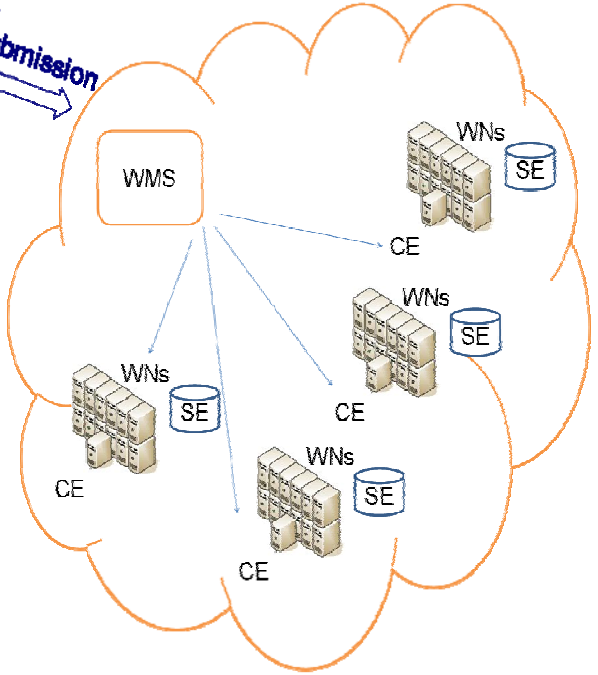
3 – 4 – 5 Job creation and submission

1) CRAB takes user code and configuration file
2) CRAB uses the dataset name to query the Data Discovery System and to find out where the data are stored
3) Creates the wrapper of user analysis code to be run in the distributed environment and the package of user code
4) Splits the analysis payload according to data location
5) CRAB submits created jobs to the Grid. The SE name is used to drive the Workload Management System to match remote resources

Other CRAB functionalities:

- Monitoring of job status
- Kill and resubmission of jobs
- Retrieval and handling of user output: copy to UI or to a generic Storage Element

- These functionalities are fully authomatized using CRAB server

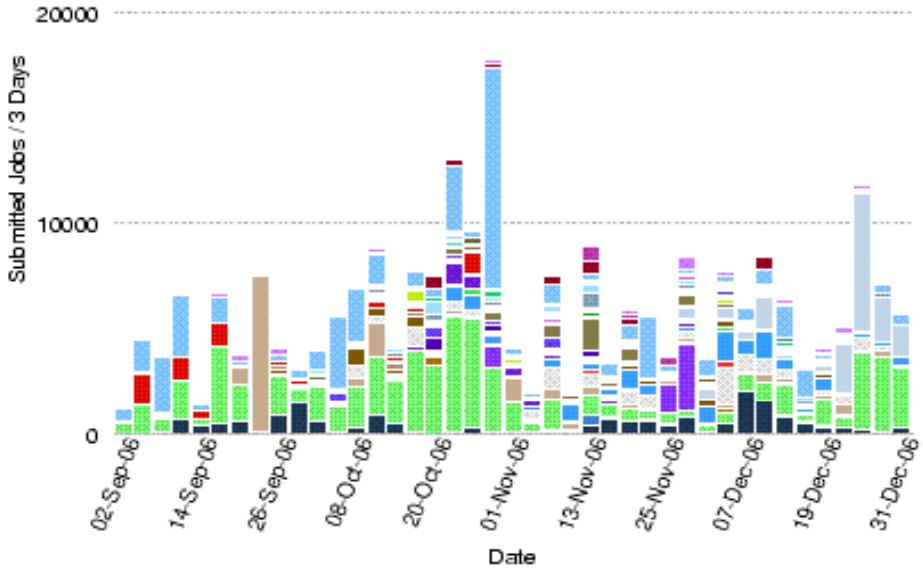


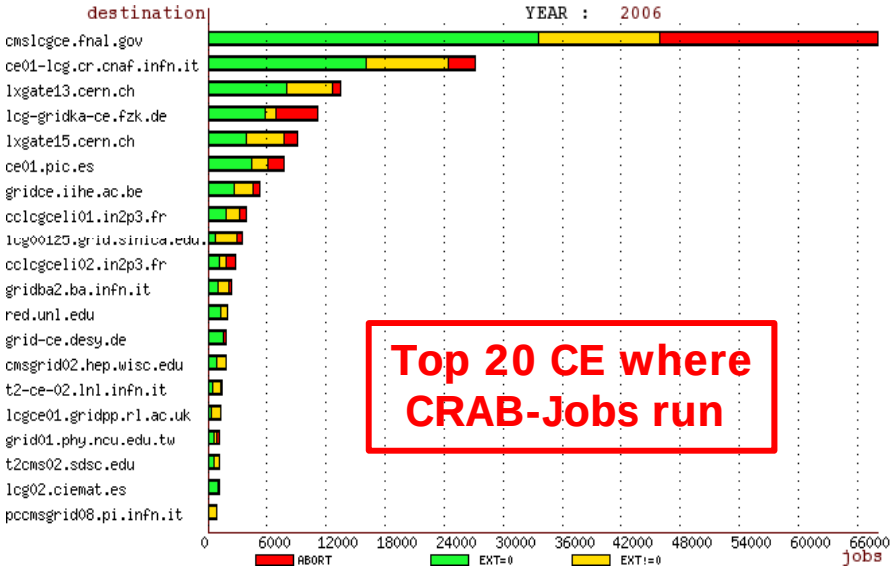
Other info

- **CRAB was born in April 2005**
- **It is written in python**
- **Users appreciated the tool, providing feedback and asking for improvements**
- **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 (CSA06) Challenges over millions of simulated events

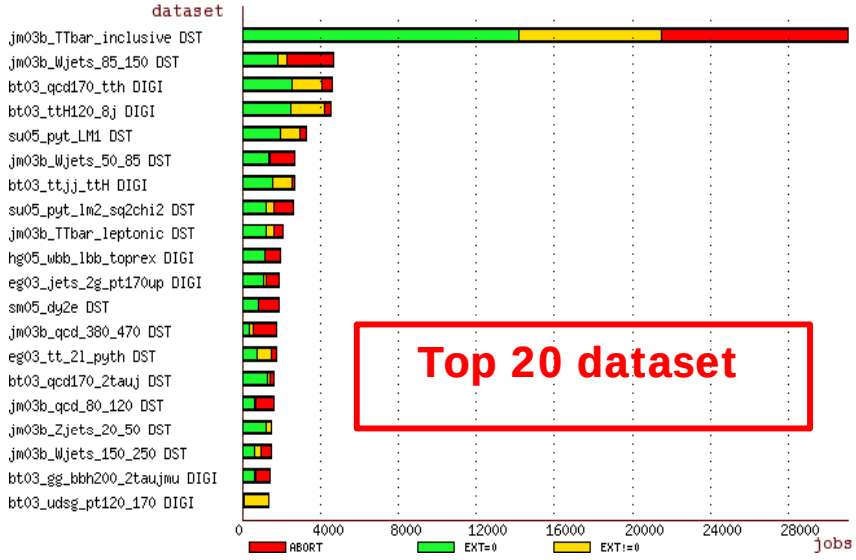
CRAB usage during CSA06

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





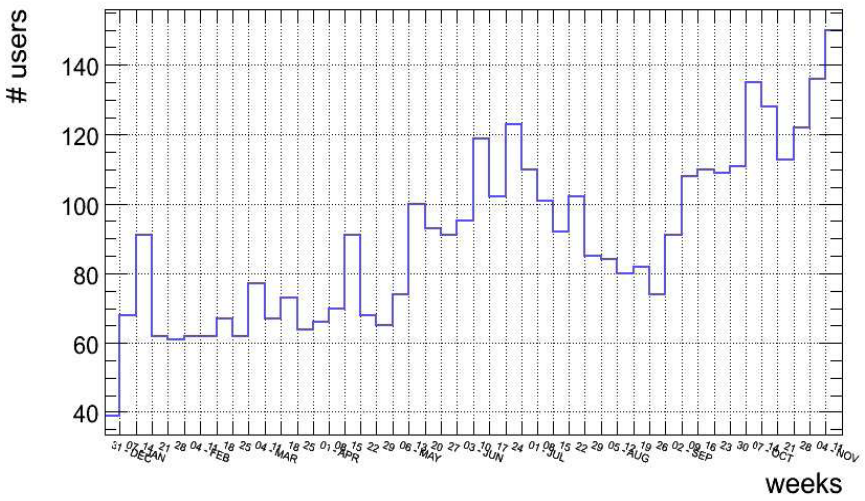
Top 20 CE where CRAB-Jobs run



Top 20 dataset

The CRAB client – server idea

- **Current users community**
  - ~600 users in 2007
  - The average daily number ~70
- **Their number is increasing**
- **The submission jobs too…**



- **When real data will be available the foreseen number of analysis job is 100 kjobs/day**
- **We need to improve the scalability of whole system.** → **CRAB client – server architecture**

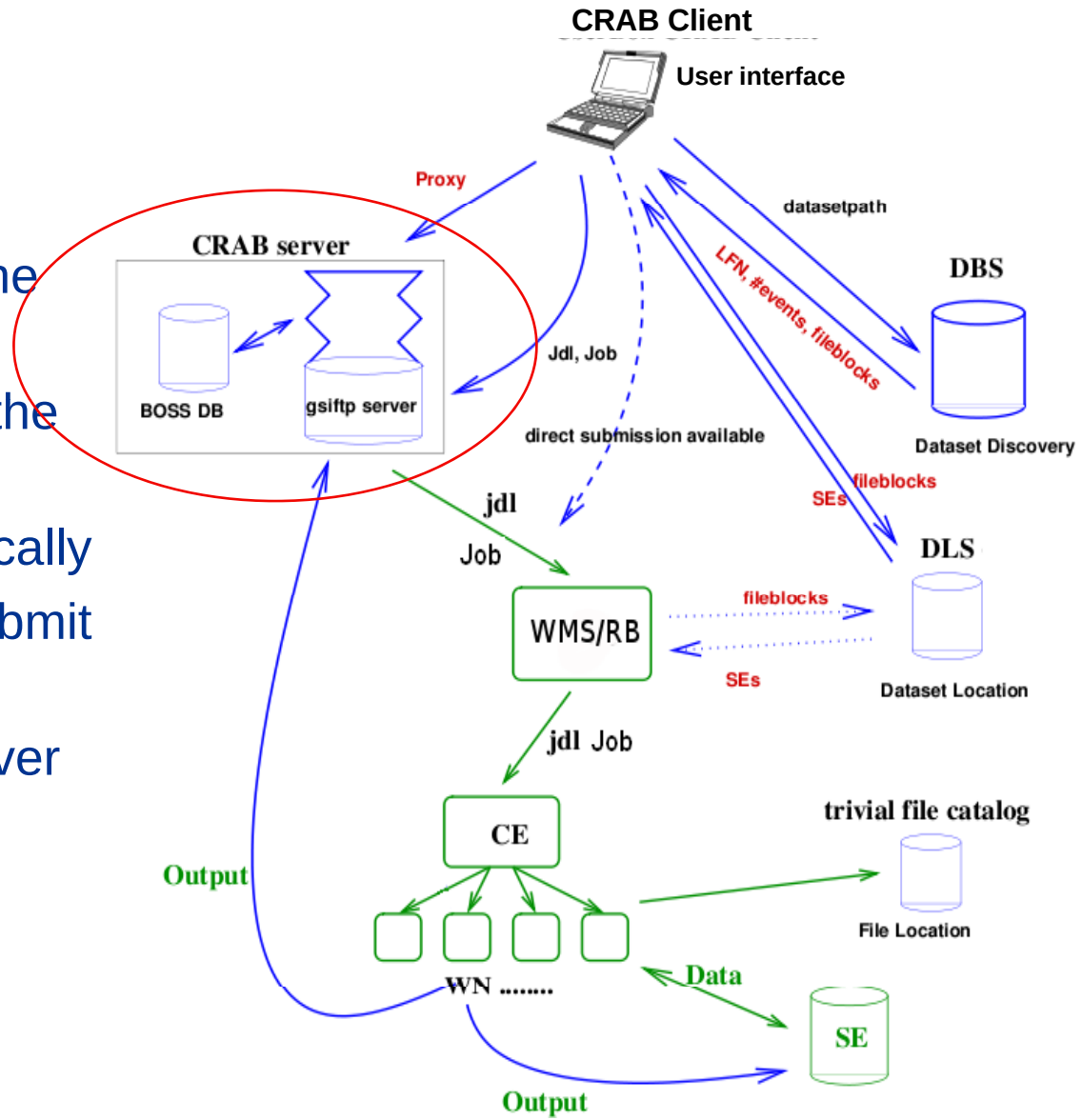
Why a CRAB server?

- **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

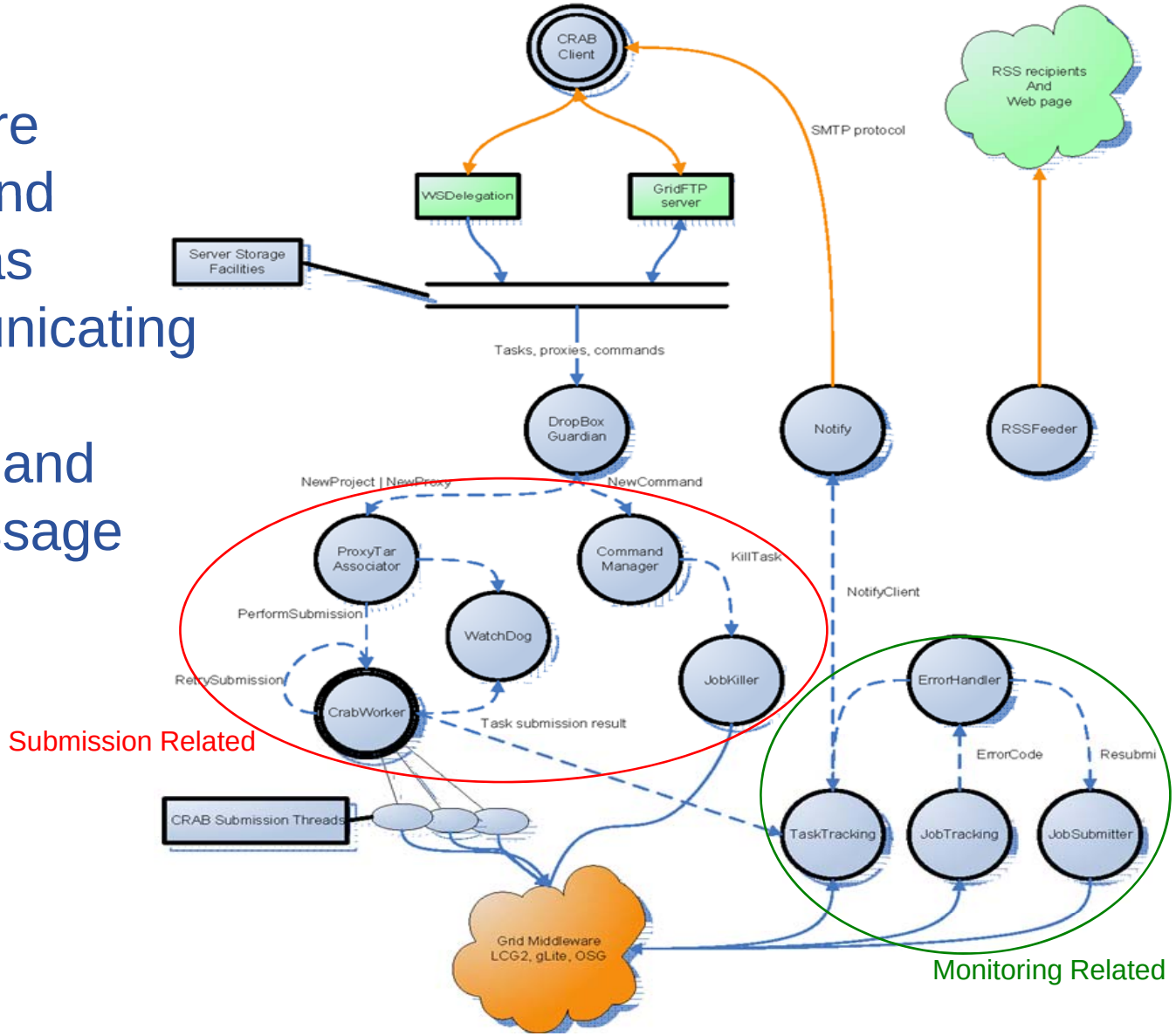
CRAB Client - Server flow

- **The server:**
  - Get tasks/credentials
  - Multi-threaded submit
  - Monitor automatically the jobs life-cycle
  - Retrieve automatically the output
  - Handle errors automatically
  - Decide whether to resubmit jobs
  - Notify the users whenever done

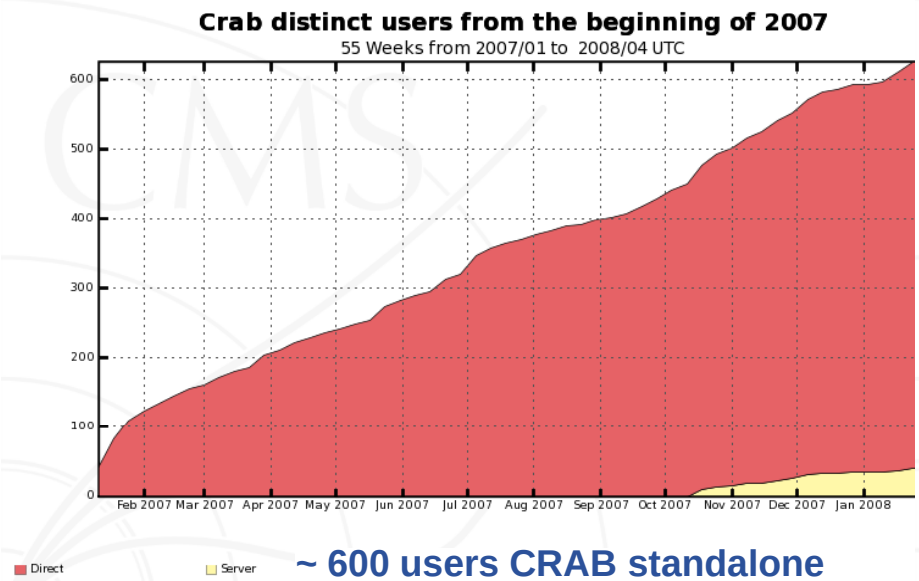


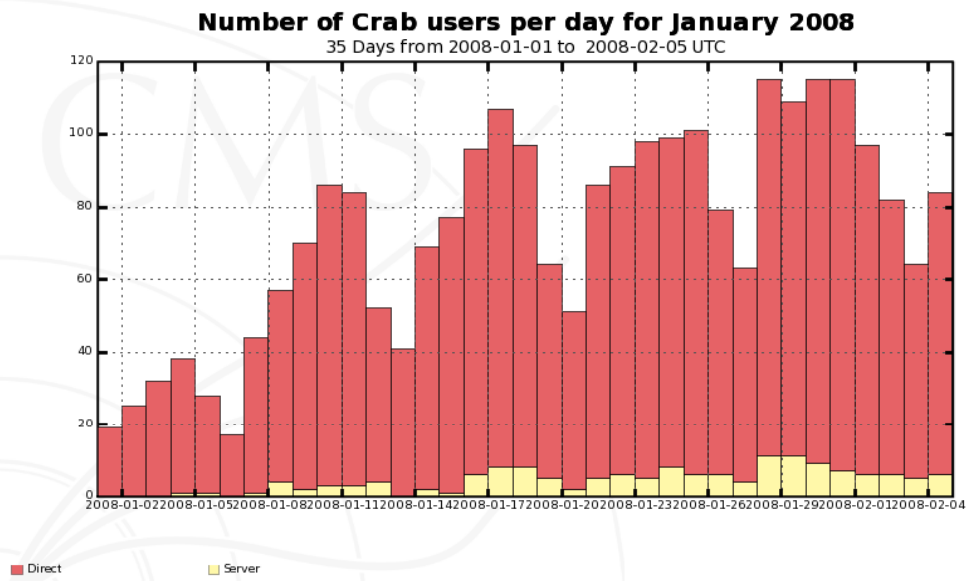
Server architecture

- The server components are independent and implemented as agents communicating through an asynchronous and persistent message service.

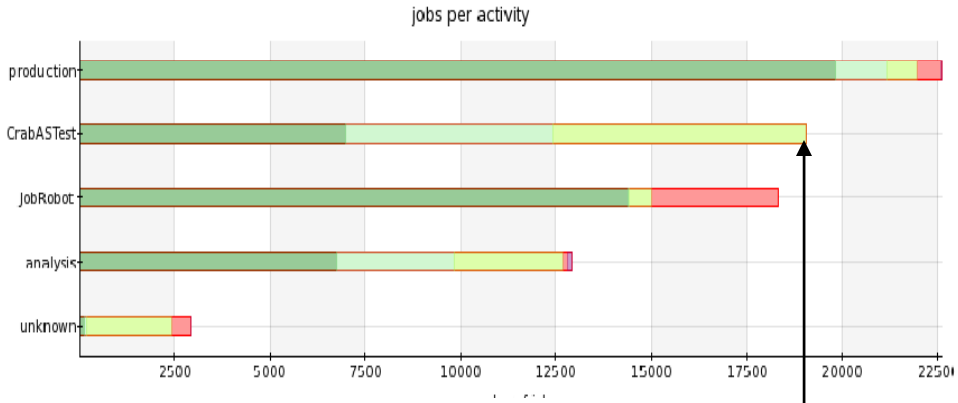


Server usage

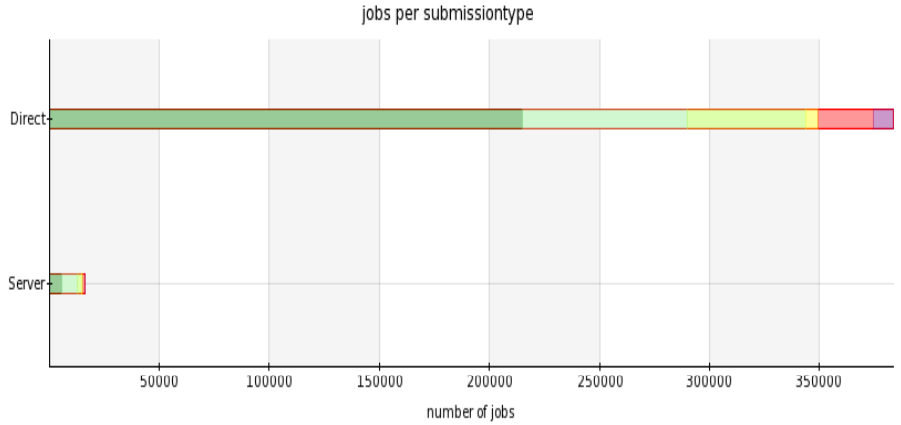




- ~ 600 users CRAB standalone
- ~ 40 users CRAB client server

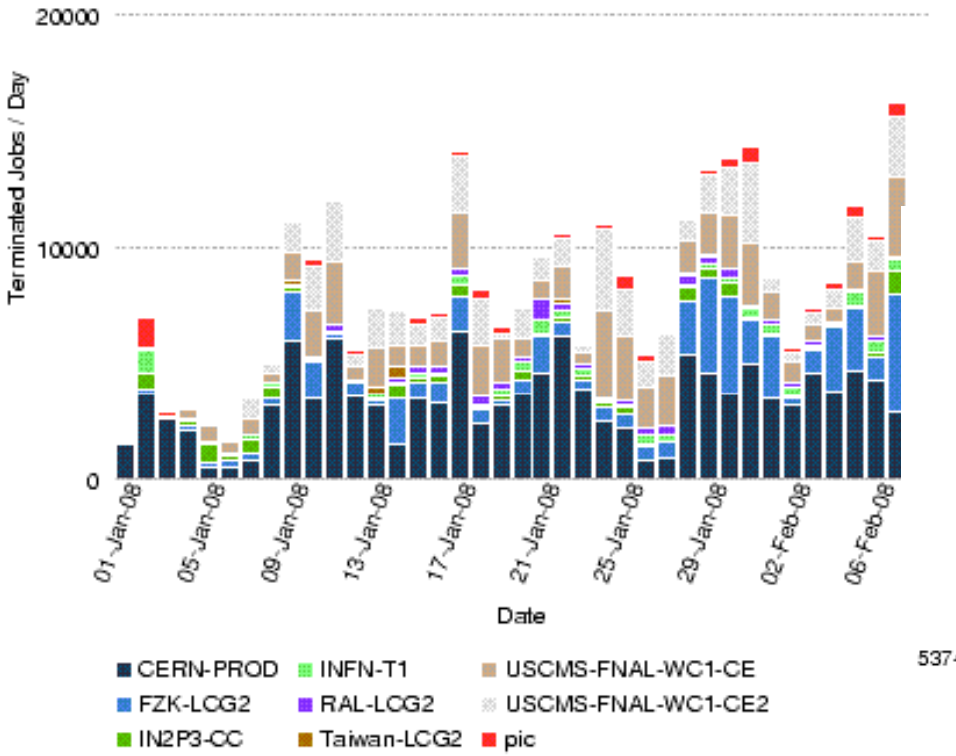


- server test 31.09-01.10 2007
- reached up to 19 kjobs in 1 day

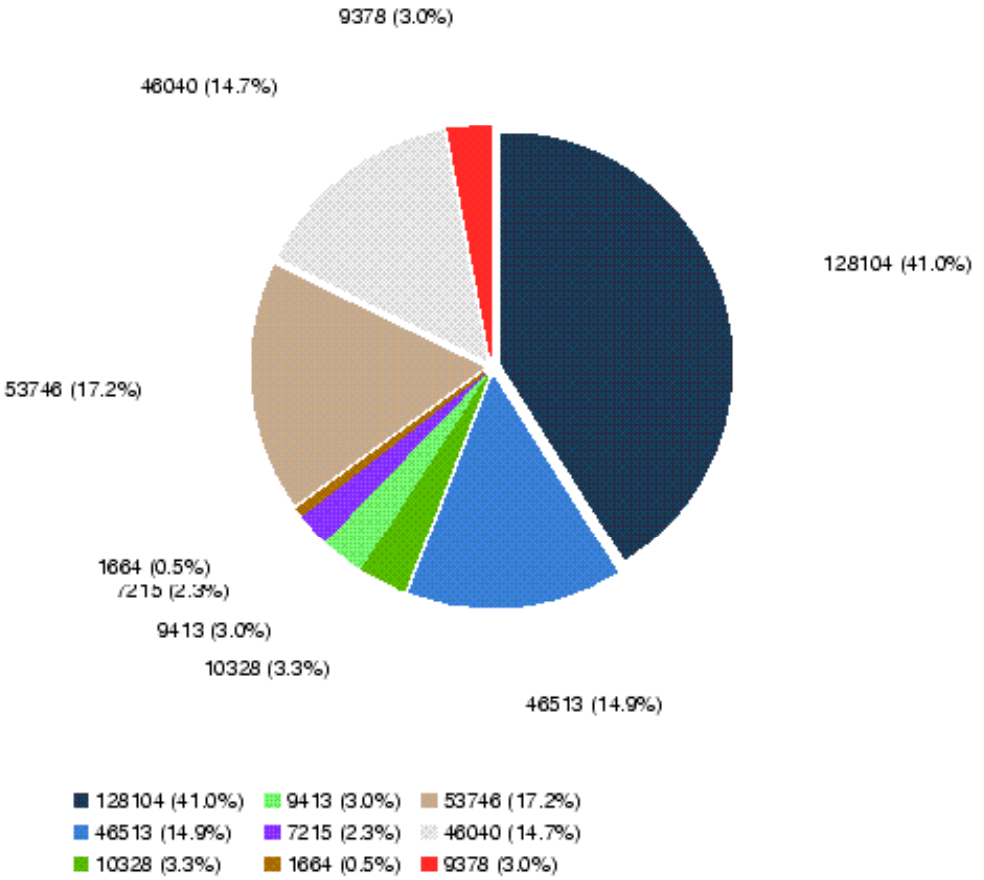


- CRAB job submitted in January 2008

Some statistics



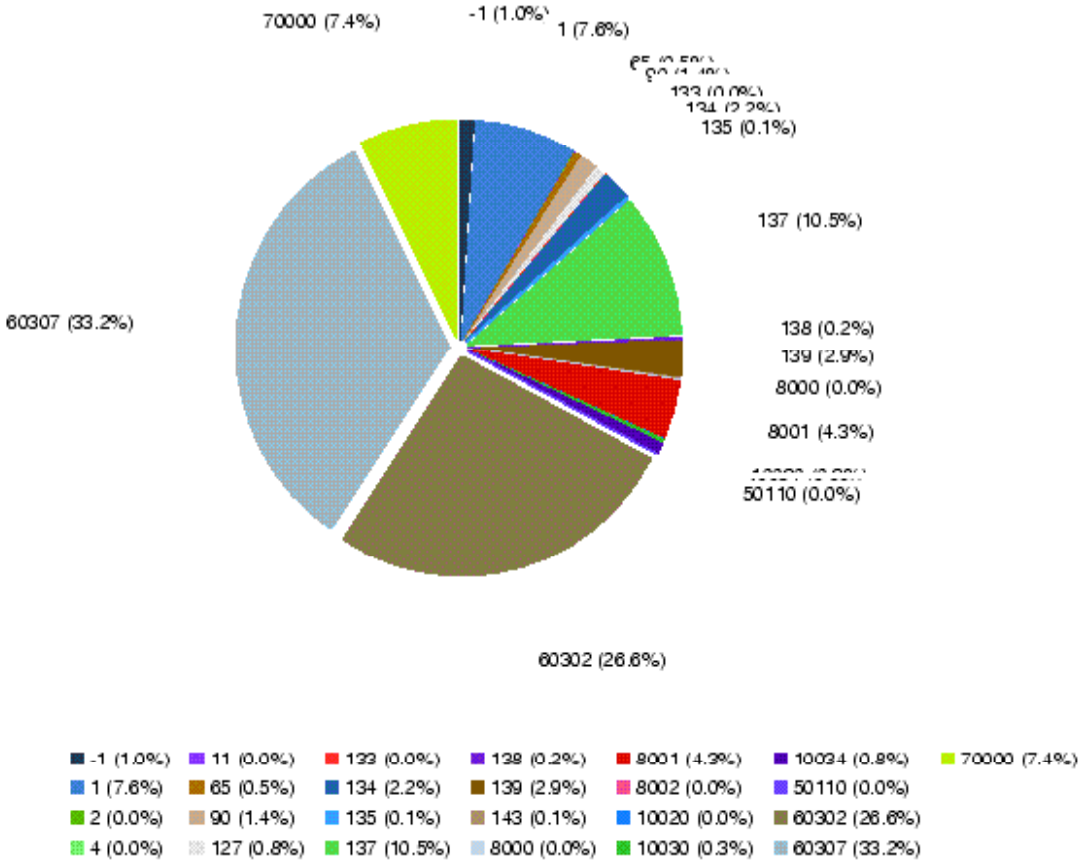
- **~300000 jobs terminated in January 2008**



Job failures

- **Grid and CMS infrastructure**
  - congestion of RB/WMS.
  - problem with the copy of output file to Storage Element
  - 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**

Failure statistics



%	Error Code	Description
33	60307	Failed to copy an output file to the SE
26.6	60302	Output file(s) not found
10.5	137	killed, unblockable (POSIX)
7.6	1	Hangup
7.4	70000	Output_sandbox too big for WMS
4.3	8001	CMSSW exception
2.9	139	Abort(ANSI)
1.4	90	Application exception
1.0	-1	Error without spec.
0.8	10034	Required application version is not found at the site
0.8	127	Error while loading shared library
0.5	65	End of job from user application
0.3	10030	VO_CMS_SW_DIR not defined

- Also due to input file not found
- Also due to problem with copy to 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
- The support is an important task: need effort by us, remote site manager and Grid service in order to improve the stability of the system
- 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**
  - http://cmsdoc.cern.ch/cms/ccs/wm/www/Crab/
- **CRAB twiki**
  - https://twiki.cern.ch/twiki/bin/view/CMS/CRAB
  - https://twiki.cern.ch/twiki/bin/view/CMS/CrabServer
- **CRAB mailing list for users**
  - https://hypernews.cern.ch/HyperNews/CMS/get/crabFeedback.html