



CRAB

CMS Remote Analysis Builder

Daniele Spiga
INFN & University of Perugia
On Behalf of CMS Collaboration

International Conference on Computing in
High Energy and Nuclear Physics
2-7 September 2007, Victoria BC, Canada

Outline

- ✓ *Introduction*
- ✓ *What is CRAB*
- ✓ *User Analysis model*
- ✓ *CRAB WorkFlow*
- ✓ *Functionalities*
- ✓ *CRABSEVER*
- ✓ *Usage*
- ✓ *Conclusions*



Introduction

CMS computing model is based on Grid

- CMS uses a tier-structured computing model and use Grid middleware to provide more computing power and optimize the use of resources for the CMS collaboration
- Data collected by the CMS online data acquisition system is sent to the Tier-0, distributed amongst T1 centers and further skimmed to associated T2 centers which provide resources for physics analysis
- The computing model foresees that all CMS users must use the Grid in order to perform its own analysis.

This talk summarize the implementation of the CMS specific tool for Grid submission and handling of analysis jobs



CRAB

CMS Remote Analysis Builder

User oriented tools which have the aim to provide an user friendly interface for CMS physicists interaction with the Grid, with data management, middleware, remote computing element, basic monitoring functionalities, etc

CRAB support:

→ **Direct submission** to the Grid

- Used since summer 2005 by end user

→ **Submission via server** ← **NEW**

Under development since February 2007

The only CMS computing tool used by end user

User Analysis Model

CRAB transports the user's analysis code to the sites where sample of interest is located, following the CMS data location driven computing model

- **User runs interactively on small samples** in the local environment in order to develop his analysis code and test it
- **Once ready the user selects a large (whole) sample** to submit the very same code to analyze xillions of events (no remote user code compilation)
- **The results are made available to the user** to be analyzed interactively to produce the final plot



CRAB WorkFlow

Job Creation:

interaction with DM system: data discovery and location
interaction with user local working area: packaging users code
set up the jobs: jobs splitting, wrapper creation...

Direct submission

Submission: Send task to Grid

Status: check Jobs/task status using Logging & Bookkeeping

Getoutput: retrieve the user output to Sandbox/SE

Submission via server

Get task from the user and submit on his behalf to Grid

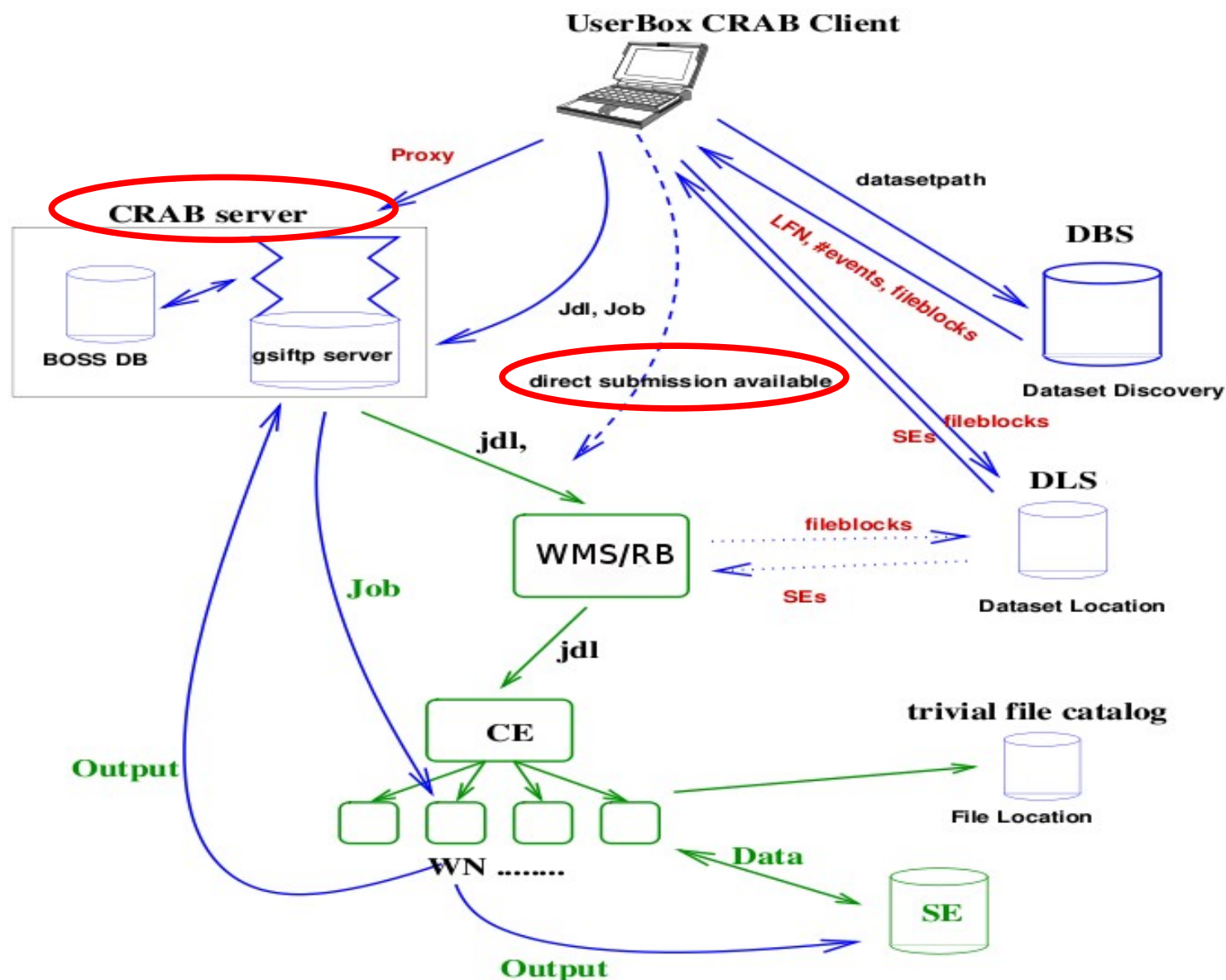
Get status of jobs/tasks and cache into local DB the informations (using BOSS)

Automatic output retrieve for completed jobs (using BOSS)

Error handler and single job resubmission if needed

Track task: collecting infos and notifies the user via mail

WorkFlow schema



Main features

- ✓ The user-CRAB interaction is performed through simple configuration file
- ✓ Live switching from standalone to server
Direct and server submission are absolutely **transparent to the user**, same configuration, command line etc.
- ✓ Main functionalities
 - Job creation, submission, status, getoutput
 - kill jobs
 - postMortem
 - job resubmission
 - task clean

CRABSEVER role

- ✓ Automatize as much as possible the interaction with the Grid, including submission, resubmission, error handling, output retrieval, etc . . .
- ✓ Try to reduce the unnecessary human load, moving all possible actions to server side, reducing to a minimum those on client side (the User Interface).
- ✓ Improve scalability of the whole system.



CRABSERVER Architecture

The server adopts a modular software approach: independent components are implemented as agents communicating through an asynchronous and persistent message service (publish & subscribe model). The core is a MySQL DB.

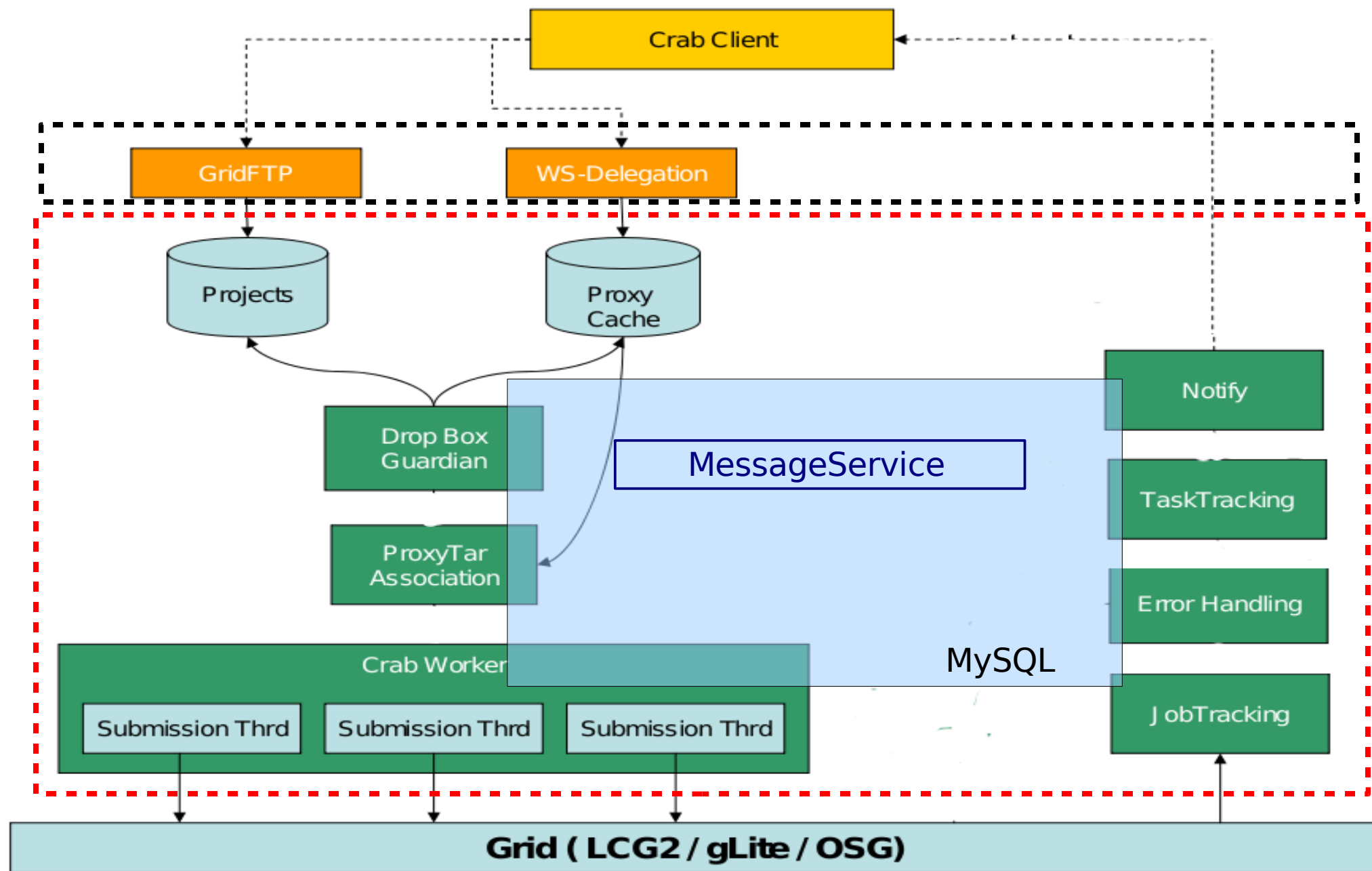
(share as much as possible the CMS production system components) see D.Evans "CMS MC Production System Development & Design" [235]

Externals:

GridFTP server: manage client-server communications

WS-Delegation service: proxy delegation service

Components



Used to access data during most of the past CMS challenges:

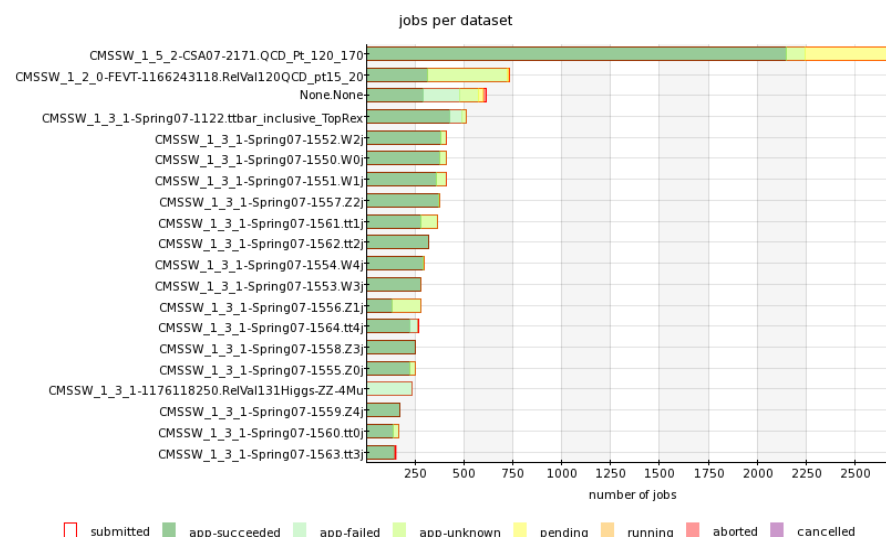
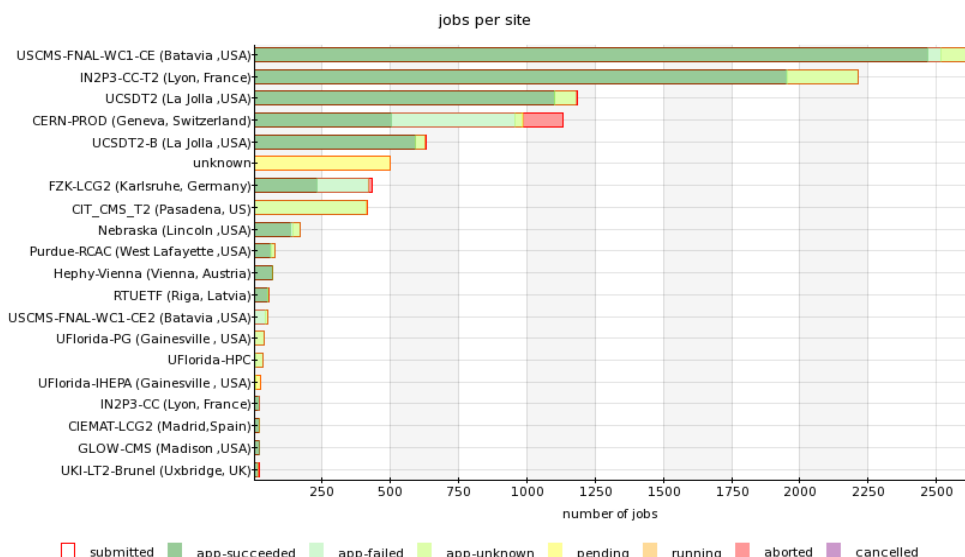
- ➔ Magnet Test Cosmic Challenge
- ➔ Tracker real data
- ➔ Physics TDR
- ➔ CSA06 on millions of simulated events

see Talk O.Gutsche

"WLCG scale testing during CMS data challenges" [240]



CMS real user analysis jobs: 02/09/07 – 03/09/07 (~24h) ~9.5K jobs





Server Usage: first results

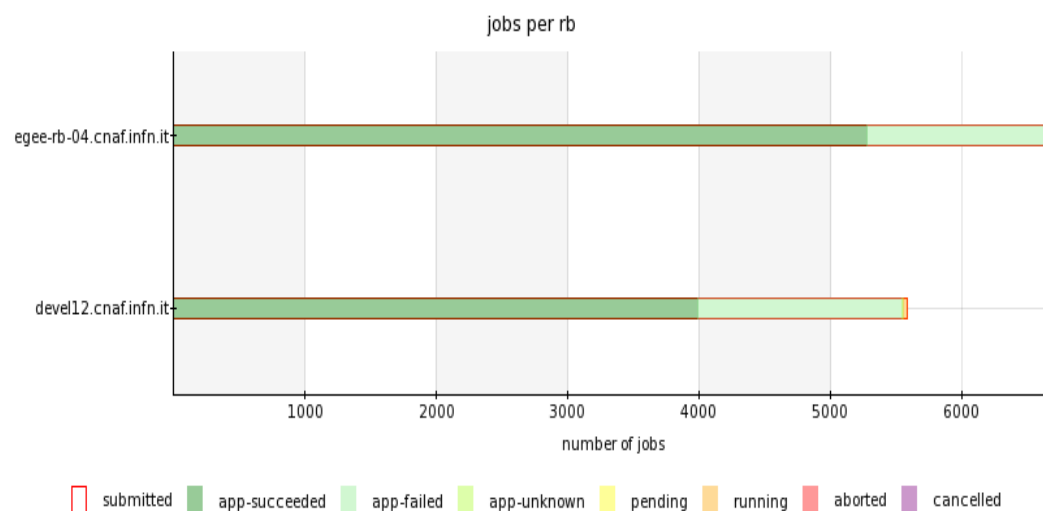
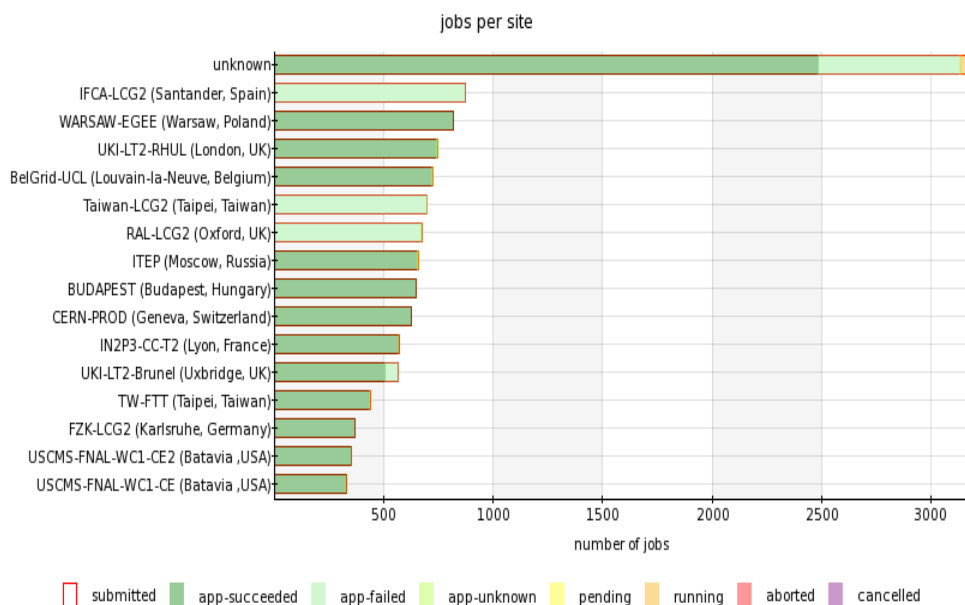
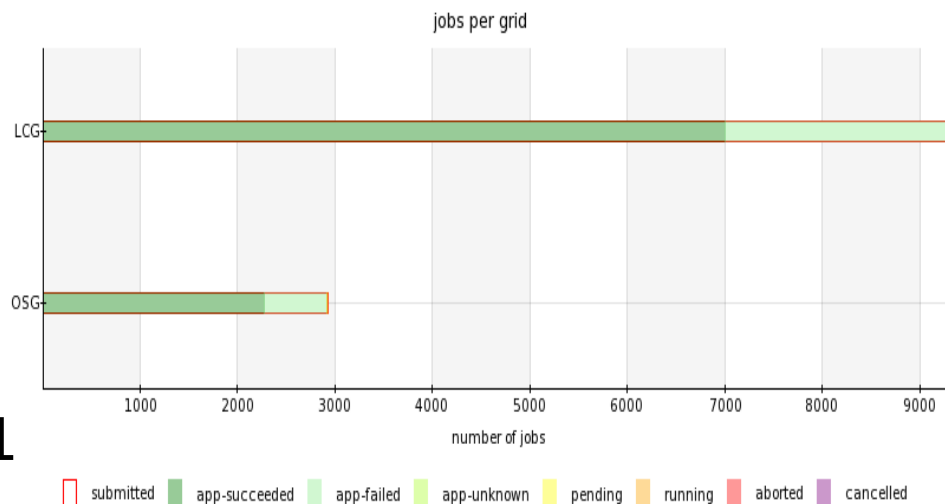
Preliminary results:

Test Run:

from 24/08/07 6:00PM
to 25/08/07 10:00AM

job submission through
CRABSERVER and gLite WMS_3.1

CMS user analysis-like jobs (~16h) ~13Kjobs



Conclusions

- From chep06 to chep07 the **CRABSERVER** has been developed and the first release is done
- Next server release under preparation
- Complete interoperability LCG – OSG
- Supported and maintained both direct and server submission
- CRAB is used with success by hundreds of users
- CRAB was used to access data during most of the past CMS challenges and will be used for coming CSA07
- Hard work waiting first real data on July 2008