WLCG scale testing during CMS data challenges

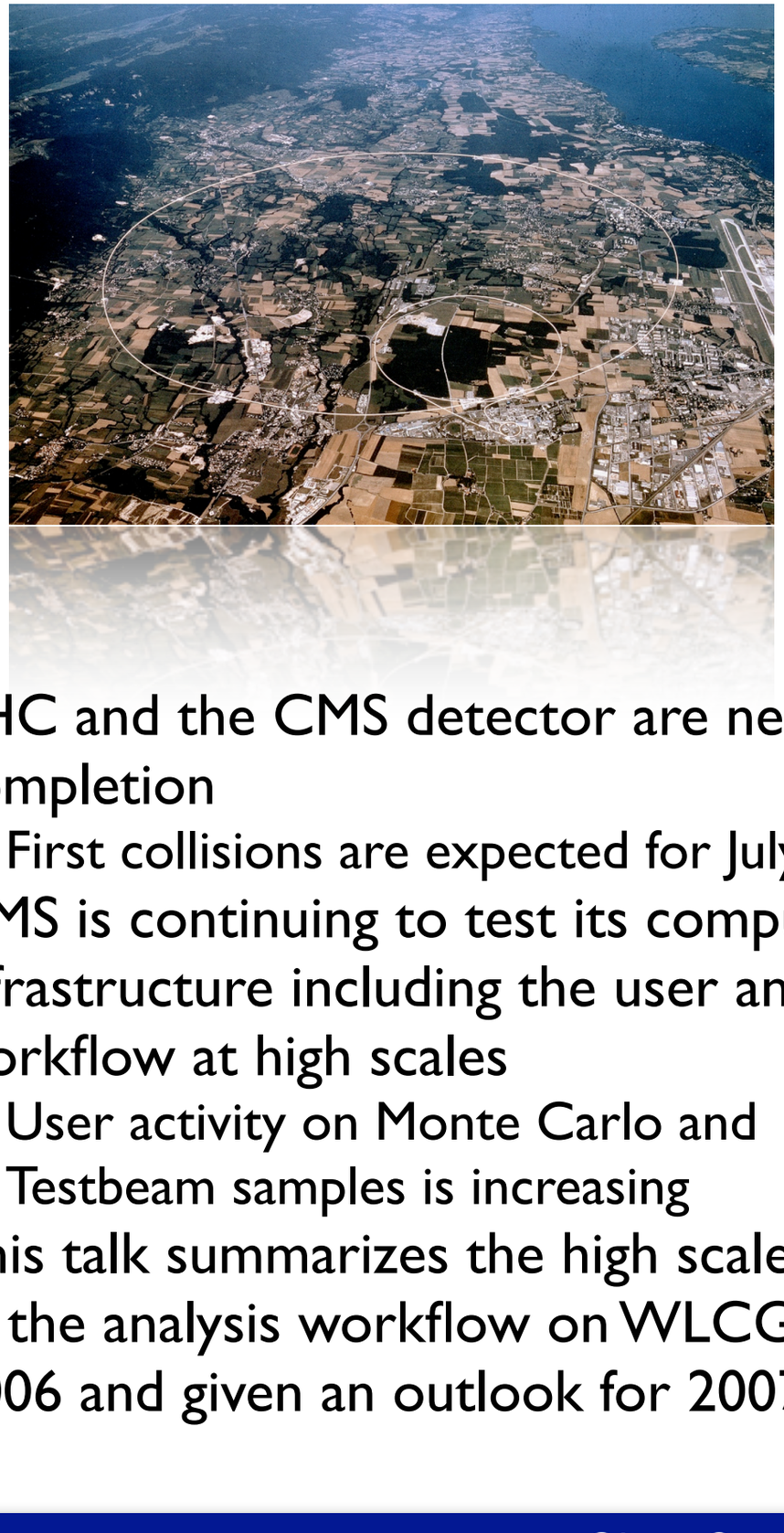
Grid middleware and tools: GM 1
Abstract 240

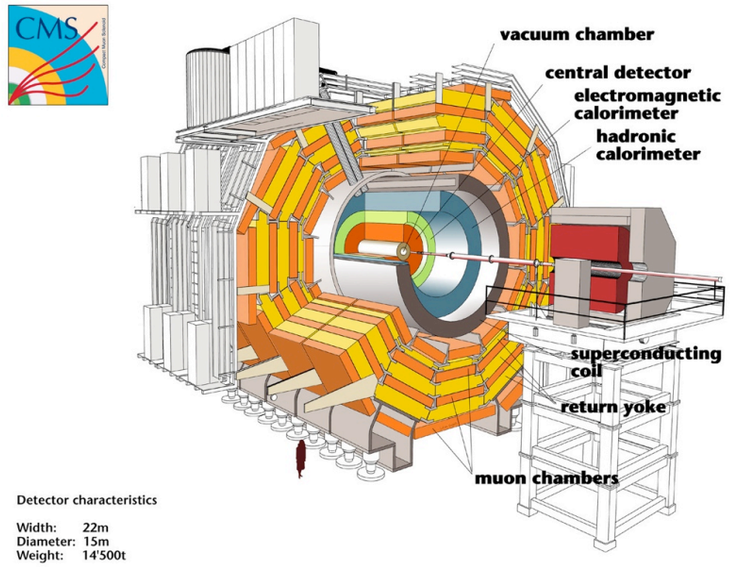
International Conference on Computing in High Energy and Nuclear Physics

2-7 September 2007, Victoria BC, Canada

Csaba Hajdu (KFKI Budapest, Hungary)
Oliver Gutsche (CMS / Fermilab)

Introduction





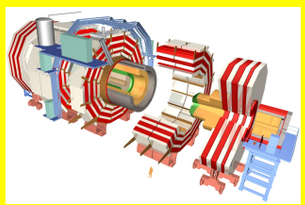
- LHC and the CMS detector are nearing completion
  - First collisions are expected for July 2008
- CMS is continuing to test its computing infrastructure including the user analysis workflow at high scales
  - User activity on Monte Carlo and Testbeam samples is increasing
- This talk summarizes the high scale tests of the analysis workflow on WLCG in 2006 and given an outlook for 2007

Outline

- CMS computing model
- CMS user analysis tool CRAB
- CMS computing challenge 2006
  - Goals
  - Automated submission infrastructure
  - Tuning and results
- Summary and Outlook

CMS computing model

- ~2000 physicists scattered around the globe want to analyze CMS data
- Analysis is location driven ("Job is sent where the data is stored.")
- CMS follows GRID approach to distribute data storage and processing world-wide using WLCG



CERN

- Data recording
- Primary reconstruction
- Partial Reprocessing
- First archive copy of the raw data (cold)

USA, **UK**, **Italy**, **France**, **GER**, **Spain**, **Taiwan**

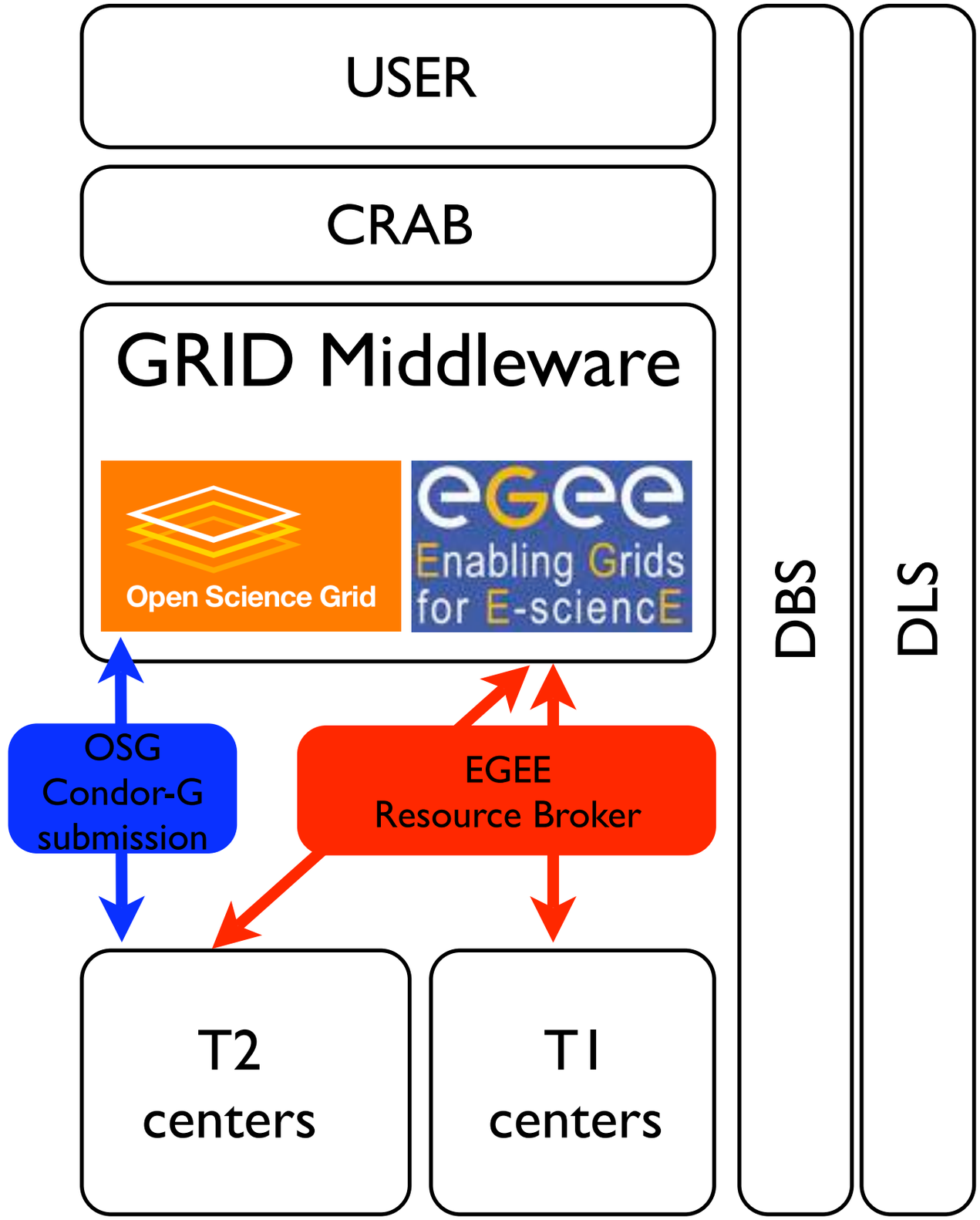
- Share of raw & reconstructed data for custodial storage
- Data Reprocessing
- Analysis Tasks (Skimming)
- Data Serving to Tier-2 centers for analysis
- Archive Simulation

- Monte Carlo Production
- **Primary Analysis Facilities**

1 Tier 0 — 7 Tier 1 — 25-50 Tier 2

CMS user analysis

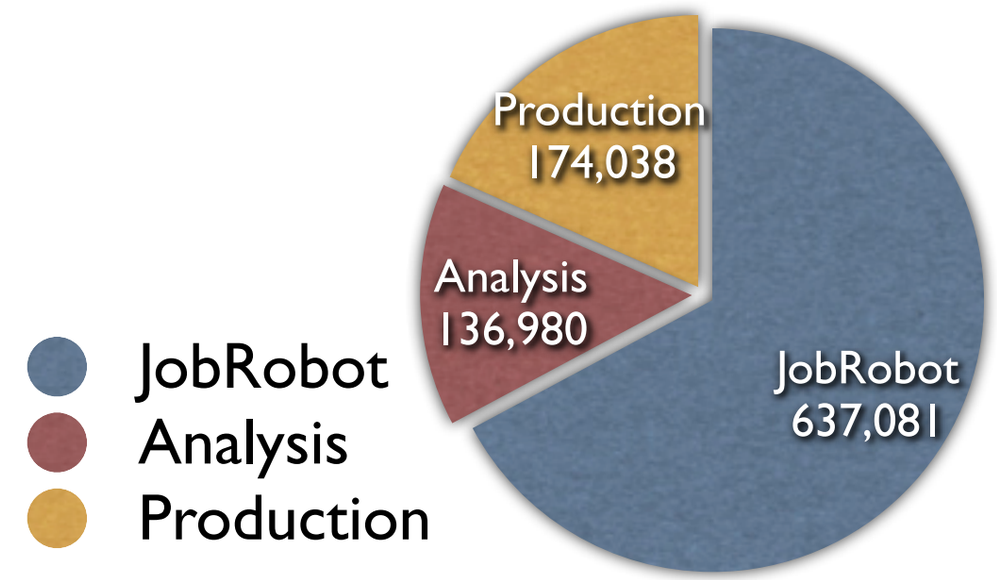
- CMS requires equal fair share access to all CMS data for all CMS users
- User tool: **CRAB**
  - **C**MS **R**emote **A**nalysis **B**uilder
    - 4 simple user steps
      1. Job Creation including data discovery and job splitting
      2. Job submission via LCG/gLite RB or Condor-G
      3. Job status check
      4. Job output retrieval
    - See also: *CRAB (CMS Remote Analysis Builder)*, Abstract 314 in track "Distributed data analysis and information management", Thursday, 09/06/07, 5:50 PM
  - External components:
    - BOSS: CMS bookkeeping service to manage job information on local UI
      - Default: SQLite (MySQL possible)
    - CMS Data Discovery Services:
      - Data Bookkeeping Service (DBS)
        - See also: *The CMS Dataset Bookkeeping Service*, Abstract 325 in track "Software components, tools and databases", Monday, 09/03/07, 3:40 PM
      - Dataset Location Service (DLS)



CMS computing challenge 2006

- **CMS Computing, Software and Analysis Challenge 2006 (CSA06)**
  - 02. October - 15. November 2006 (6 weeks)
  - Test of data flow and data handling of CMS computing model
  - 25% capacity test of what is needed for operations in 2008
- Tested components:
  - Prompt reconstruction at T0
    - Data distribution to T1
  - Calibration, re-reconstruction and skimming at T1
    - Data distribution to T2
  - **Analysis jobs at T2**
    - and also T1

- **Goal:**
  - ~**50,000 jobs per day** to exercise the job submission infrastructure
    - ~**10,000 jobs per day** exercising skimming and re-reconstruction at T1 (central operations / production)
    - ~**40,000 jobs per day** analysis jobs, combination of
      - user submitted jobs
      - robot submitted analysis-like jobs
- Total number of submitted jobs during CSA06: 948,099



Automated test infrastructure: JobRobot

- Purpose:
  - Simulate user analysis by repetitive submission of analysis-like CRAB jobs
- Implementation:
  - Perl agents executing CRAB commands (create, submit, check status, getoutput)
  - Dummy analysis job using the CMS software framework CMSW (read in data and print into logfile)
- Requirements:
  - Continuously send jobs to all published datasets
  - Sustain constant job rate at individual centers and avoid burst submission

TaskSource

- 1 agent:
  - Queries catalog (DBS/DLS) for available datasets at all sites
  - Per site and dataset:
    - Check for currently running and pending jobs at the site
    - Prepare CRAB configuration (user choice before using CRAB)
    - Put project in TaskPrepare queue

TaskPrepare

- Several agents:
  - Per queue entry:
    - Execute CRAB creation step using project configuration
    - Use CRAB data discovery (DBS/DLS)
    - If creation is successful, put project in TaskSubmit queue

TaskSubmit

- Several agents:
  - Per queue entry:
    - Execute CRAB submission step using available resources check from RB
    - If no resources available (site fails site availablility tests) discard queue entry
    - If submission is successful, put project in TaskQuery queue

TaskQuery

- Several agents:
  - Per queue entry:
    - Monitor progress using CRAB status check
    - Cleanup aborted jobs
    - If all jobs have succeeded or failed, put project in TaskCollect queue

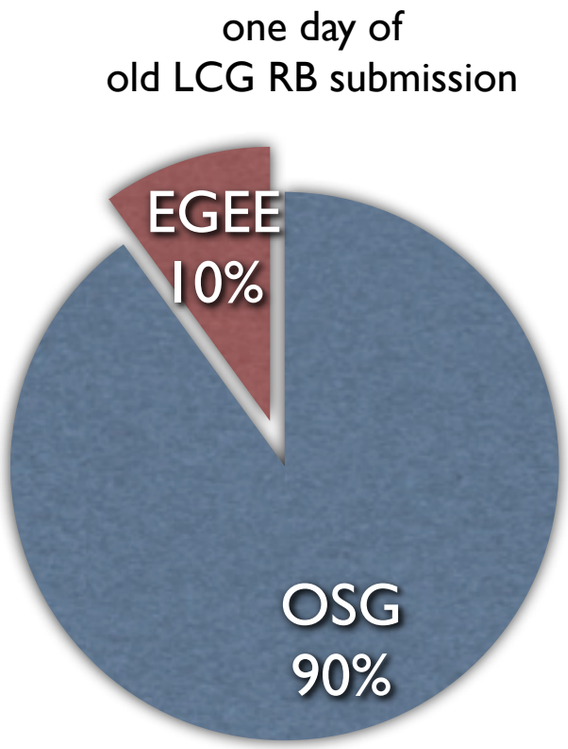
TaskCollect

- Several agents:
  - Per queue entry:
    - Collect log files and output using CRAB output collection
    - Cleanup project

Timeline & Observations

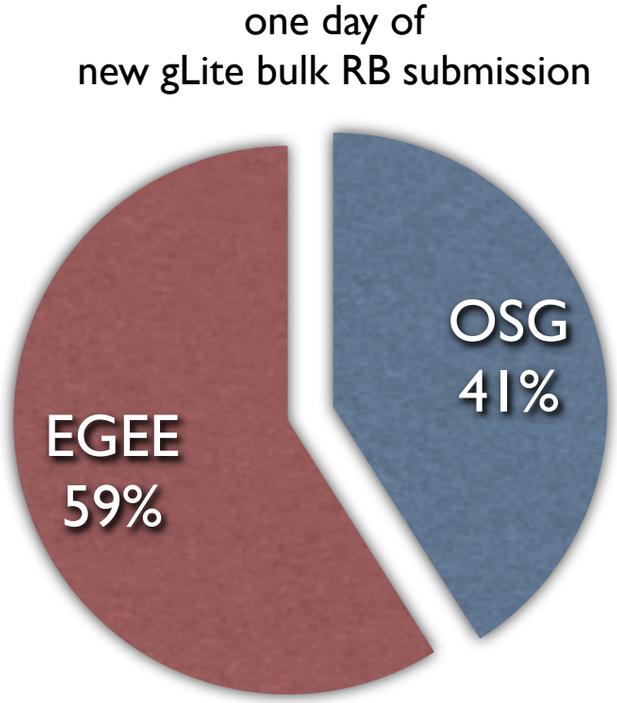
start

- October 15
  - JobRobots started analysis submission at 10,000 jobs/day
  - 2 JobRobot instances on separate machines:
    - Robot 1 using the old EDG RB submitting to LCG sites
    - Robot 2 submitting to OSG sites using Condor-G
  - Both JobRobots use standard JobRobot and CRAB infrastructure



gLite bulk submission

- October 16,17
  - CRAB integrated bulk submission to new gLite RB
  - EGEE fixed a bug in the gLite UI within 24 hours



Quill

- October 19,20
  - Scaling issues on Condor-G robot
    - Continuously job status queries overload the local running Condor scheduler under high load (several thousand jobs in the queue)
    - Use external Quill Postgres DB to improve query behavior

Timeline & Observations

SQLite / MySQL

limit bulk submission

BOSS MySQL fix

- October 24,25
  - Scaling issues with continuous CRAB status checks using local SQLite DB's of BOSS (high I/O load on machines)
    - Effect: too many jobs submitted to single CE's because the status check was not updating the number of running and pending jobs at a site (several killed CE's)
    - Moved to central MySQL server to decrease I/O load
  - Scaling issue with gLite RB: projects with more than 1000-2000 jobs introduce very low submission efficiencies
    - Introduced limit of submitted jobs per CRAB project
  - Timeout problem of central MySQL server for BOSS
    - Installed second MySQL server, one for each robot
    - Traced back to incomplete MySQL query and promptly fixed by BOSS developer team

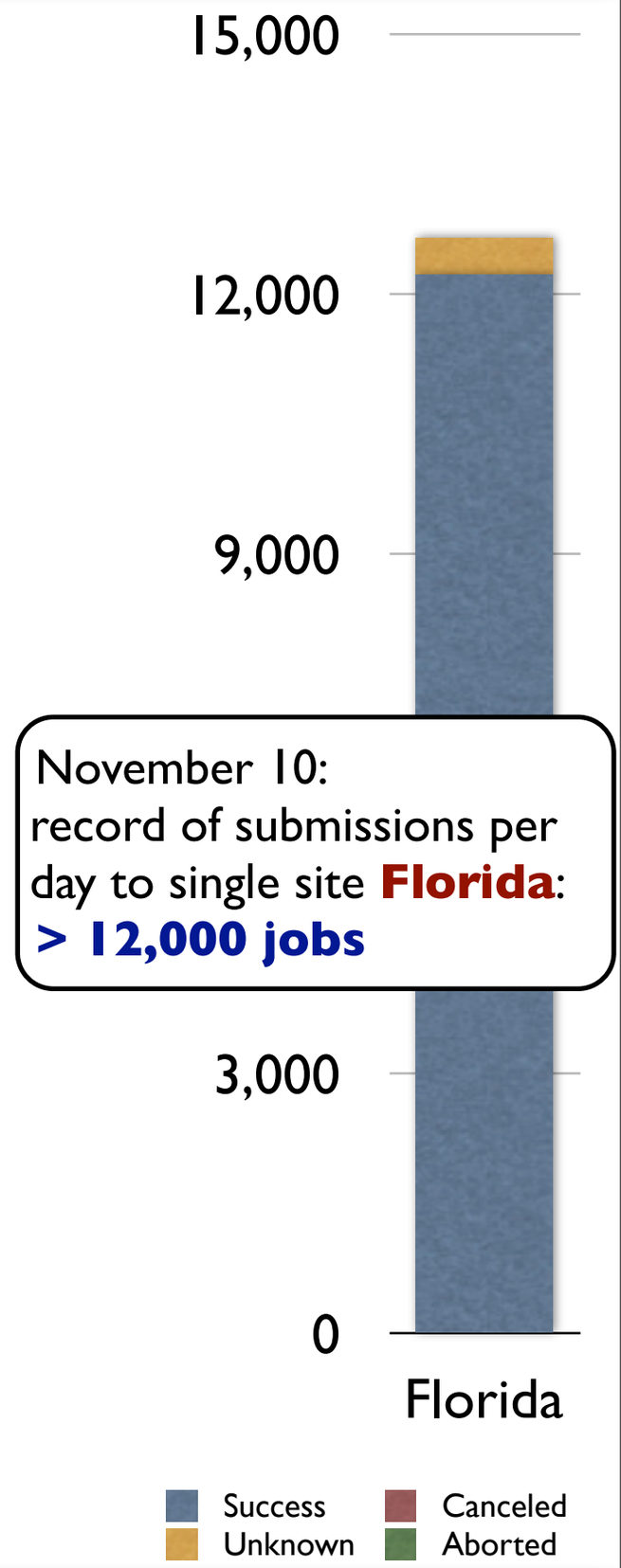
Timeline & Observations

2 robot instances maxed out

11 robot instances

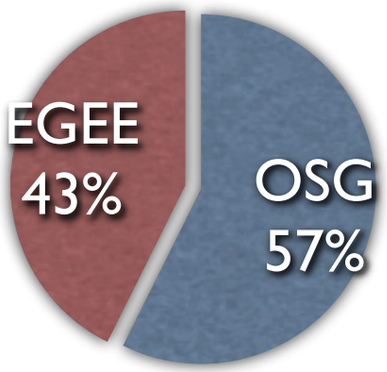
27 robot instances

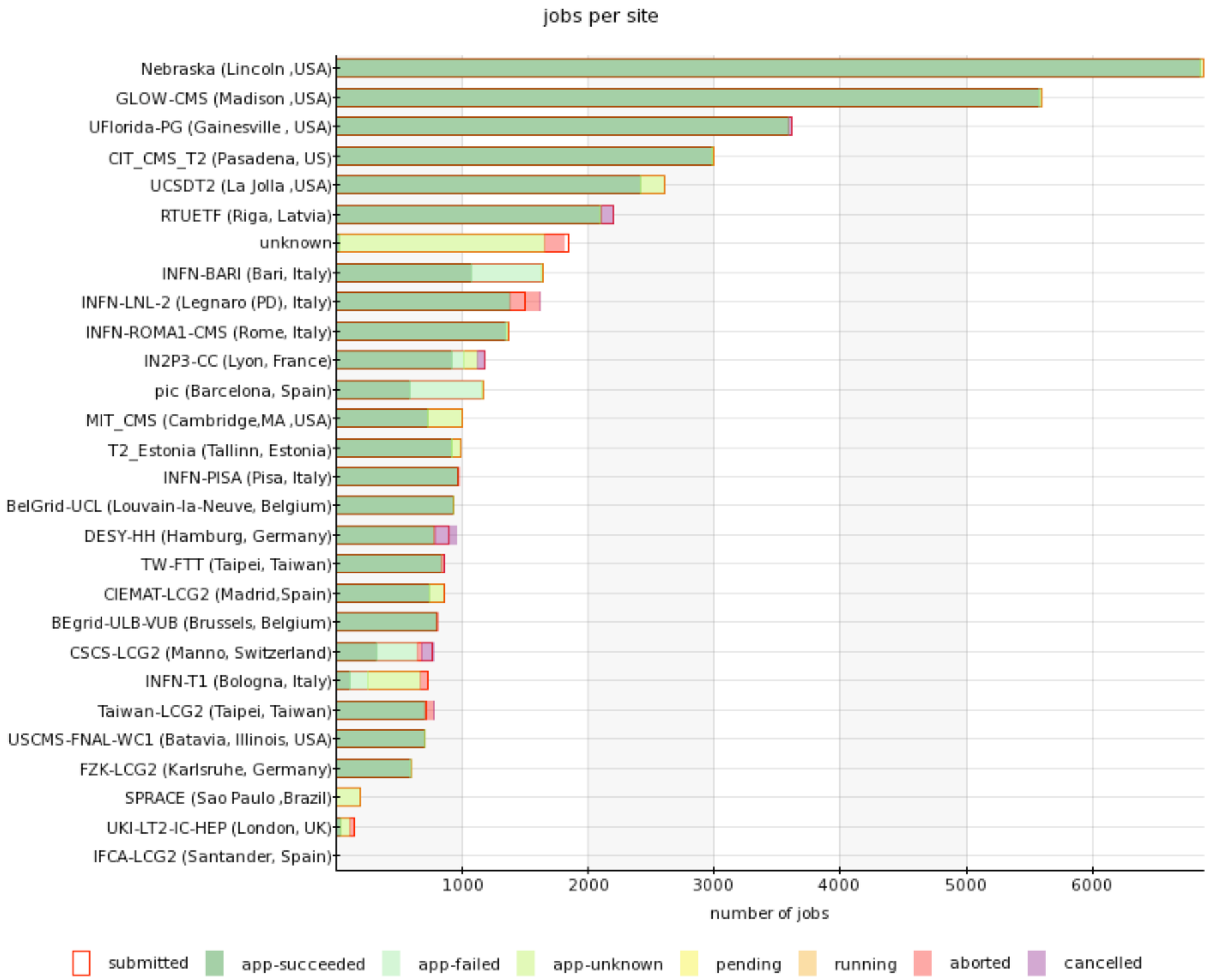
- October 31
  - Current JobRobot deployment not sufficient to reach required scale
  - Concurrent agent operations on the same machine slow down the JobRobot operation
    - I/O access to various logfiles
    - DB access to MySQL of BOSS and Postgres of Condor
  - Move to more robot instances installed on different machines (currently 11 including the original 2), the new robot instances:
    - Moved back to SQLite for BOSS
    - Moved back to local Condor schedulers without Quill Postgres DB's
  - Job rate went up to 20,000 - 25,000 jobs per day
- November 5, 6
  - Original 2 robots showed again limitations by sending more jobs to more sites (inclusion of all T1 sites)
  - More robots were deployed (27 in total)
  - Job rate went up to 40,000 - 45,000 jobs per day



CSA06 goal reached

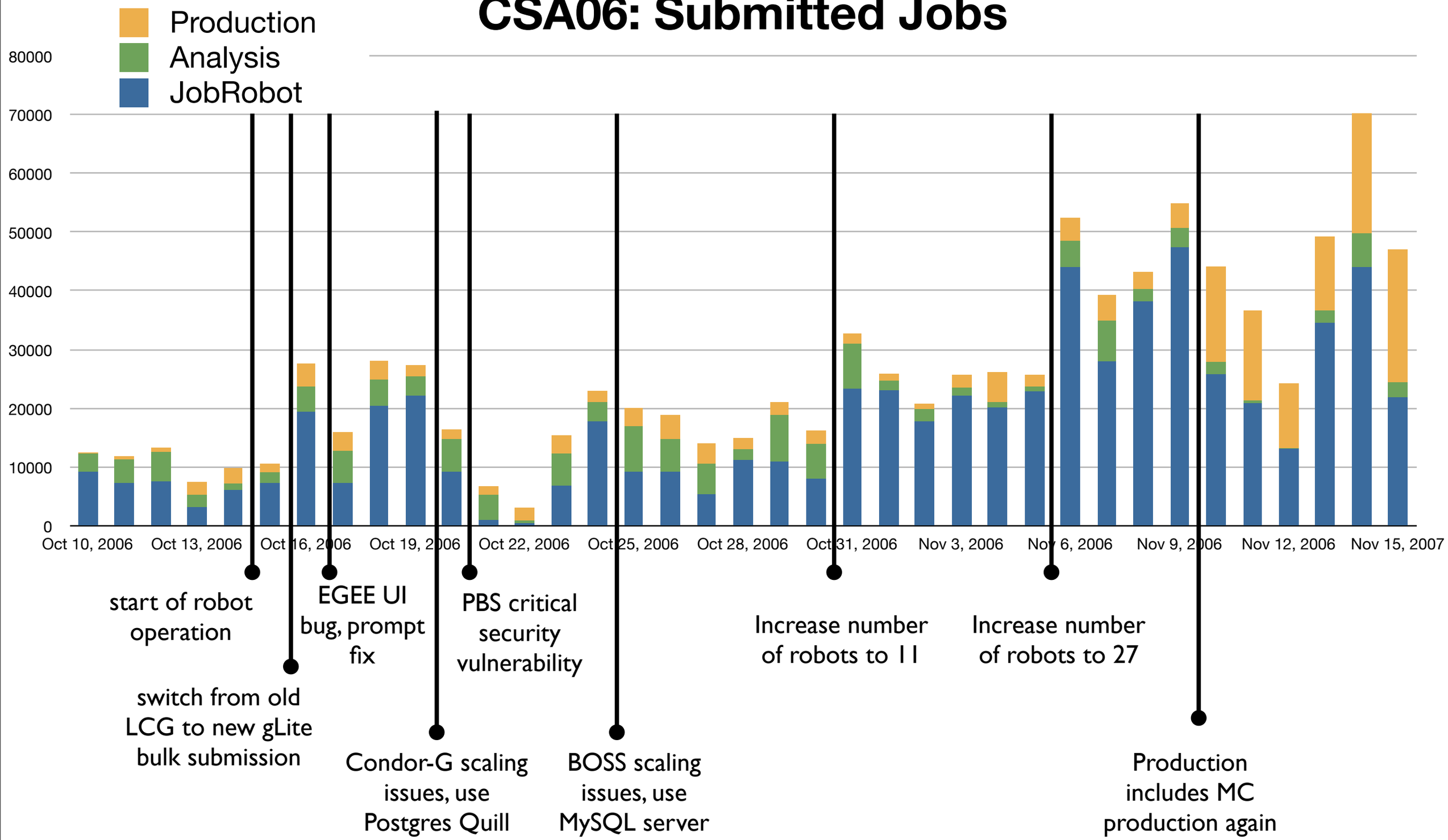
- November 6
  - Combined robot, production and analysis submission exceeds 50,000 jobs per day
  - 27 robots submitting 44,000 successful jobs/ day
  - Condor-G and gLite bulk submission used
    - gLite bulk submission is using 3 different RB's at CERN
    - Distribution follows resource availability on the two GRIDs





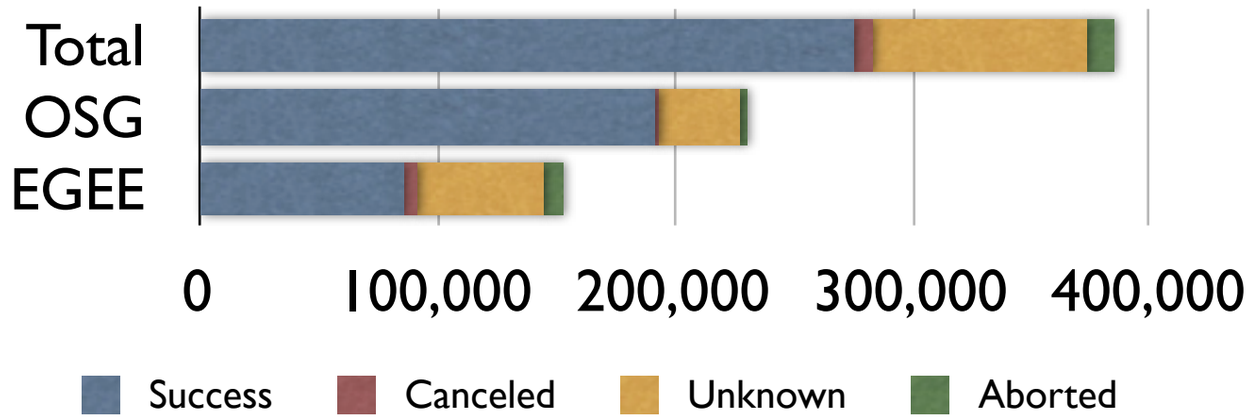
Timeline

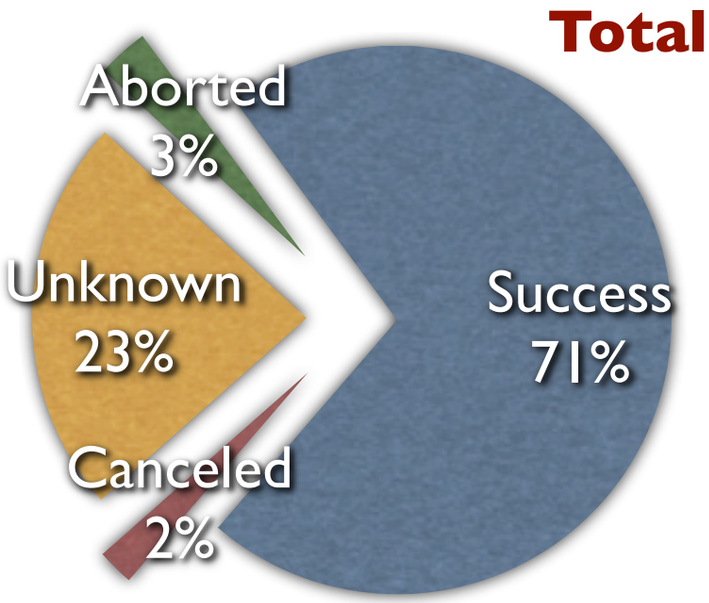
CSA06: Submitted Jobs

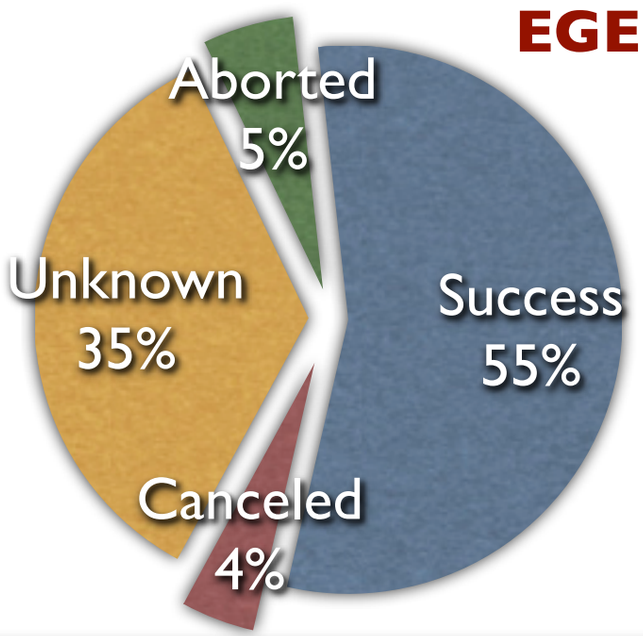


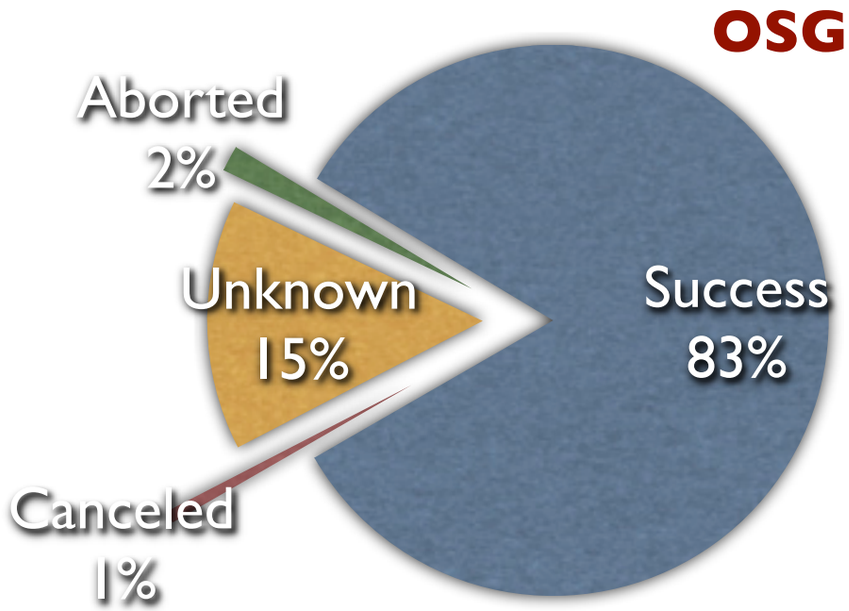
GRID efficiency

- Success rate in timespan with highest job submission scale between October 31 to November 10 from the GRID point of view
  - **Success**: GRID submission including status check and output retrieval succeeded
  - **Canceled**: GRID job was submitted but canceled by the robots due to problems or too long pending times
  - **Unknown**: Problems in the monitoring (monitoring information didn't or incompletely reached the collection service) or jobs never reached the WN (GRID problems, monitoring can pickup GRID flavor when job starts on WN, not before)
  - **Aborted**: GRID job was aborted by the GRID middleware









Summary & Outlook

Summary

- Goal of the job submission part of CSA: 50,000 jobs/day
- Two central submission instances were not able to reach required scale also after significant tuning effort
- Multiple robot instances plus user and production submissions reached goal using both EGEE (gLite bulk submission) and OSG (Condor-G submission) resources
- GRID efficiency good at high scales (many improvements for EGEE already implemented and in operation)

Outlook

- CMS will conduct a next challenge (CSA07) in 2007 at 50% scale of needed capacity for 2008
- Job submission goal will be 100,000 jobs/day
- Due to growing user base and increased physics activities, all jobs are expected to be submitted by users with a small percentage of central production effort
- Challenging test for the infrastructure due to chaotic nature of user submission compared to automated robot operation