The CMS Workload Management

Daniele Spiga

University and INFN of Perugia

On behalf of the CMS Collaboration

Innovative Particle and Radiation Detectors

1 - 5 October, 2006

Siena, Italy
Outline

✓ Monte Carlo production system introduction
✓ ProdAgent
✓ User Analysis introduction
✓ CRAB (CMS Remote Analysis Builder)
✓ Conclusion and Plans
Introduction

- CMS uses a tier-structured computing model and use grid middleware to optimize distributed resources providing computing for the CMS collaboration [see talk Daniele Bonacorsi]

- The different computing centers use two Grid middleware implementations:
  - LHC Computing Grid (LCG) / Enabling Grids for E-sciencE (EGEE)
  - OpenScience Grid (OSG)

- The CMS Workload Management system deals with this structure covering two main different purpose:
  - Monte Carlo production (ProdAgent)
  - User analysis (CRAB)
Monte Carlo Production

Monte Carlo (MC) production is crucial for detector studies and physics analysis.

- The whole complex workflow of the MC Production consists of multiple steps:
  - Generation → Simulation → Digitisation → Reconstruction

- MC event generation is performed in the distributed grid environment using Tier 2 resources.

The tool which take care of the MC Production System performing all the steps of its workflow is the ProdAgent.
The main objective is the automatization of the production task:

- Manage requests from users
- Break requests into jobs
- Perform submission and tracking of jobs
- Handle errors and perform job resubmission
- Keep track of data output in DLS/DBS

Data Management system allows to discover, access and transfer event data in a distributed computing environment. Data Bookkeeping (DBS) and Data Location system (DLS) track respectively which data exists and where data is located.

- Perform merge of output files
- injecte data into database for the transfers by using Phedex

The CMS tool which take care of data transfer to reach destination site
General Architecture

ProdManager (*ProdMgr*) (and the policy piece) manage the assignment of requests to one or more ProdAgents and tracks the global completion of the task.

ProdAgent Requests work allocations from ProdMgr (next slide)
ProdAgent Plugin structure

Components are implemented as independent agents. Communication is performed through an asynchronous and persistent message service.

- Automatic Job preparation + Job submission via BOSS
- Job Tracking (Bookkeeping) based on BOSS (Batch Object Submission System) provides an easy to use book keeping system.
- Automatic interactions with local scope DBS/DLS
  - Lookup DBS/DLS
  - Register produced data in DBS/DLS
- Merge step: merge small outputs files produced by individual jobs in fewer larger files
ProdAgent Usage

Used mostly during the preparation of MC simulated datasets

For Computing Software and Analysis Challenge 2006 (CSA06)

Which is a test at 25% of the CMS distributed computing capacity needed in 2008

[see talk Nicola De Filippis]

From 1 July 2006

- 66M event produced following all the production steps

- 9 Physics sample ready (Minimum bias, T-Tbar, Z→μμ....)
Real (from next year) and simulated data are used by the physicists

- Using CMS analysis framework (CMSSW) the physicists develop physics selection code to be run on (distributed) data
- Data are distributed amongst T1 centers and further skimmed to associated T2 centers
- CMS user must interact with the grid to access distributed dataset

**CRAB** provides the user with a framework to run users analysis code on the grid
The main goal of the project is to provide an user friendly front-end for interaction with the grid for CMS, including interaction with data management, middleware, remote computing element, basic monitoring functionalities, etc.

Users have just to develop their analysis code in an interactive environment and then interact with CRAB via cfg file specifying:

- Analysis code with possibly parameter cards
- Which dataset need to run on
- How to split the jobs
- Output files name and how to manage them
CRAB job flow

**CRAB Takes care of:**

- **Input data discovery**  
  (querying DBS and DLS)

- **Job creation**  
  - job splitting according with user requests  
  - packaging of user code (cards)  
  - wrapper to run on worker node  
    - set proper environment  
    - run user code  
  - jdl (xml) creation

- **Job submission to LCG WMS (RB)**  
  (via BOSS)

- **Monitoring of job status and output retrieval**

- **Handling of user output**  
  copy to UI or a generic Storage Element

**Plus a sets of other minor functionality:**  
kill job, resubmit, postmortem...
CRAB status

- Developed to support the old CMS Framework (ORCA) and usable also with other job type (FAMOS)

- Is now fully working with the new CMS software CMSSW

- Inter-operability with LCG - OSG support submission via Resource Broker to LCG and OSG sites transparent to final user.
CRAB Usage

Reached more than 100'000 jobs/month daily record is about 10'000 jobs.

Several tens of users distributed over the world

Top 20 used User Interface
CRAB usage (past experience)

Based on old CMS Framework (ORCA)

Top 20 used CE and dataset

Each bar represents the total number of jobs and it is divided into three categories:
- jobs that produce user executable - Exit Code equal to 0
- Exit Status different from 0
- jobs that could not run due to the grid problems

About 300 different dataset have been accessed
CRAB jobs can run on about 45 different Computing Element
Conclusion

ProdAgent

- New tool in production from last 3 month
- produced successfully 66M events

CRAB

- extensively used by physicists to access data for Physics TDR
- in production since 2005
- Fully working with new framework and Data Management
Plan

- Both the tools are always under development

- Improve the integration with the Framework and Data management

- Use the lesson will lernt from the tool usage during the ongoing challenge (CSA06) to discover and solve eventual bottlenecks, in order to improve the system.