PROCEEDINGS OF SCIENCE

Large Scale Job Management and Experience in Recent Data Challenges within the LHC CMS experiment.

D. Evans, D. Mason, O. Gutsche
Fermi National Laboratory
Batavia, IL, USA
E-mail: evansde@fnal.gov, dmason@fnal.gov, gutsche@fnal.gov

S. Metson
Bristol University
Bristol, UK
E-mail: s.metson@bristol.ac.uk

S. Wakefield[1]
Imperial College London
London, UK
E-mail: stuart.wakefield@imperial.ac.uk

D. Hufnagel, A. Hassan
CERN
Geneva, Switzerland
E-mail: Dirk.Hufnagel@cern.ch, ahmad.hassan@cern.ch

A. Mohapatra
University of Wisconsin
Madison, WI, USA
E-mail: ajit@hep.wisc.edu

M. Miller
Massachusetts Institute of Technology
Cambridge, MA, USA
E-mail: mlmiller@mit.edu

F. van Lingen
California Institute of Technology
Pasadena, CA, USA
E-mail: fvlingen@caltech.edu

[1] Speaker

From its conception the job management system has been distributed to increase scalability and robustness. The system consists of several applications (called ProdAgents) which manage Monte Carlo, reconstruction and skimming jobs on collections of sites within different Grid environments (OSG, NorduGrid, LCG) and submission systems such as GlideIn, local batch, etc...

Production of simulated data in CMS mainly takes place on so called Tier2s (small to medium size computing centers) resources. Approximately ~50% of the CMS Tier2 resources are allocated to running simulation jobs. While the so-called Tier1s (medium to large size computing centers with high capacity tape storage systems) will be mainly used for skimming and reconstructing detector data. During the last one and a half years the job management system has been adapted such that it can be configured to convert Data Acquisition (DAQ) / High Level Trigger (HLT) output from the CMS detector to the CMS data format and manage the real time data stream from the experiment. Simultaneously the system has been upgraded to facilitate the increasing scale of the CMS production and adapting to the procedures used by its operators.

In this paper we discuss the current (high level) architecture of ProdAgent, the experience in using this system in computing challenges, feedback from these challenges, and future work including migration to a set of core libraries to facilitate convergence between the different data management projects within CMS that deal with analysis, simulation, and initial reconstruction of real data. This migration is important, as it will decrease the code footprint used by these projects and increase maintainability of the code base.

XII Advanced Computing and Analysis Techniques in Physics Research
Erice, Italy
3-7 November, 2008

1.CMS production and processing

During 2009 CMS will collect approximately 1200M raw events [1], in addition to this an equivalent amount of Monte Carlo simulation data will be produced for comparison and performance studies. The computing associated with this data can be divided into 2 areas: Quasi real-time processing of the collected data stream (online) and non-time critical production and processing of all data (offline). The online dataflow consists of:

- Conversion of data from the detector read out binary format to one suitable for long term storage and processing (based on ROOT [2] persistency).
- Placing events in appropriate datasets as driven by their physics content (determined by an online trigger). Together with the first step this is known as repacking.
- Prompt derivation of alignment and calibration constants (AlCa).
- Data quality monitoring designed to flag problems with the data (DQM).
- Data processing resulting in compact data formats for use in analysis (RECO and AOD).

The online processing takes place at CERN due to time constraints (all processing must be completed within 24 hours of initial data taking). The remaining computing capacity available at CERN is insufficient for the rest of CMS' computing needs so distributed computing resources are employed. Offline processing includes production of Monte Carlo and scheduled reprocessing of both real and Monte Carlo data. Reprocessing will be carried out multiple times a year, subject to the availability of software releases and improved detector understanding, resulting in updated RECO and AOD data. This opportunity will also be used to further split datasets to optimize them for analysis use.

The distributed computing resources range from large national computing centers (Tier1) to small compute farms at universities (Tier2), in keeping with this convention, the online system is also know as the Tier0. Activities are split between the types of resources (see Figure 1) with scheduled re-processing at the Tier1's and Monte Carlo production and analysis at the Tier2's. These resources, known as sites, are accessed via a variety of grid middleware and composed of a number of different storage systems.

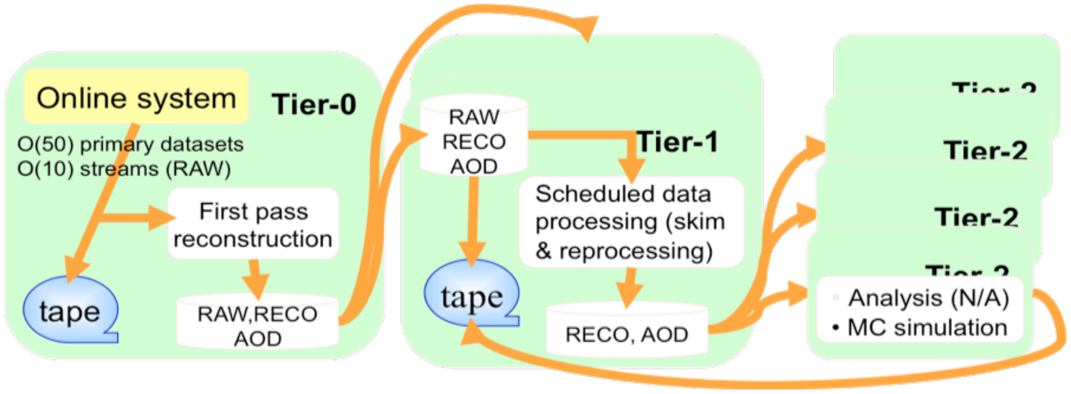

Figure 1: CMS processing activities and data flow.

2.ProdAgent

The ProdAgent software has been in use for more than 2 years. Initially developed as a tool for Monte Carlo production it has become the primary tool for all CMS's organized event processing and has recently been adopted as the basis for the online processing system. The unique requirements for online processing are enforced by an additional layer of management and accounting [3].

ProdAgent is designed to be extensible: it works with a variety of batch systems, including grid resources; multiple storage systems and a number of job and workflow types. It aims to automate as much as possible reducing the human effort required to operate it.

The generalized ProdAgent workflow is illustrated in Figure 2. The ProdAgent creates jobs as required and uses the configured submission method to run them at sites. These jobs are limited to contacting services local to the execution site to increase reliability; these services include Storage Elements (SE's) and sites' conditions data cache.

In general each job produces data products that are unsuitable for long-term storage on tape because of their small size, this limitation is overcome by merging the products from several jobs together. The intermediate data products are deleted asynchronously by subsequent jobs.

Once data has been produced at a site it is entered into the CMS file and transfer databases (DBS [4] and PhEDEx [5], respectively). It can then be transferred and made available for use at other sites. Logs are copied to the site's SE where appropriate parties can inspect them. After a grace period the logs are removed and archived on tape at CERN.

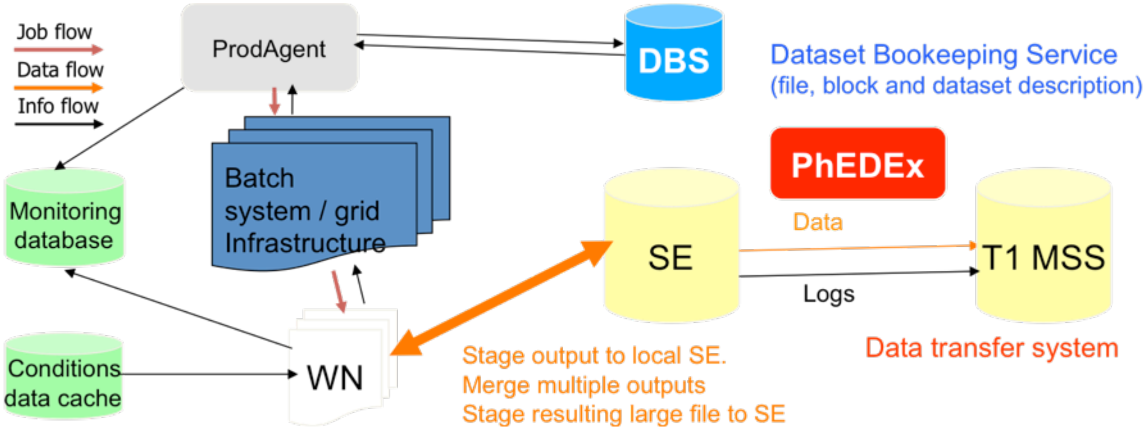
Job flow
Data flow
Info flow
ProdAgent
DBS
Dataset Bookeeping Service
(file, block and dataset description)
Monitoring database
Batch system / grid Infrastructure
PhEDEx
Data
SE
T1 MSS
Logs
Conditions data cache
WN
Stage output to local SE.
Merge multiple outputs
Stage resulting large file to SE
Data transfer system

Figure 2: Typical ProdAgent workflow

ProdAgent is composed of multiple independent components, written in Python, with a MySQL [6] database for persistency. Components are event driven and responsible for a subset of the total work required (i.e. job submission). They respond to messages by carrying out the desired work and may trigger actions in other components by sending messages.

Central to ProdAgent is the concept of a workflow, this specifies the steps (and configurations), that must be executed to complete a production or processing request. Workflows are created from a CMSSW (the CMS physics framework) configuration and associated details, such as detector conditions and output datasets, and may contain multiple processing steps. Once a workflow is given to a ProdAgent an ensemble of independent jobs are created, these are then submitted when available compute resources are identified. If data merging is required further jobs are created and submitted to perform this task. The workflow is complete when all jobs have finished and the outputs have been registered with the CMS databases listed above.

3.Recent Experiences

During 2008 a series of integration tests, termed Global Runs (GRs), were conducted as various CMS sub-detectors were completed. This culminated in a brief data-taking period with a circulating beam in September 2008. The Tier0 provided online processing for the GRs, only Repacking initially but DQM and reconstruction were active for the final runs.

Monte Carlo production and processing continued during this period, preparing data for various pre and post startup studies. As a ProdAgent instance can only be configured to use one submission mechanism CMS operated multiple instances each managing work on a subset of the resources available. Approximately 10 instances were in use during this period. Each ProdAgent instance was managed by operator(s) who were responsible for overseeing work requests and debugging problems identified during processing.

Some of these activities formed part of exercises designed to test the readiness of the computing, software and physics analyses. The most recent of these (Computing, Software and Analysis challenge of 2008, CSA08) aimed to replicate the full chain from data taking to analysis [7].

Files were fed to the Tier0, processed and datasets distributed to remote sites. Once files arrived at the Tier1s' they were reconstructed to produce RECO data, which was then transferred to Tier2s' where multiple analyses performed. The files that were fed to the Tier0 were produced before the challenge and during this period (termed CCRC08) ProdAgent reached its nominal startup goal of 100M events a month.

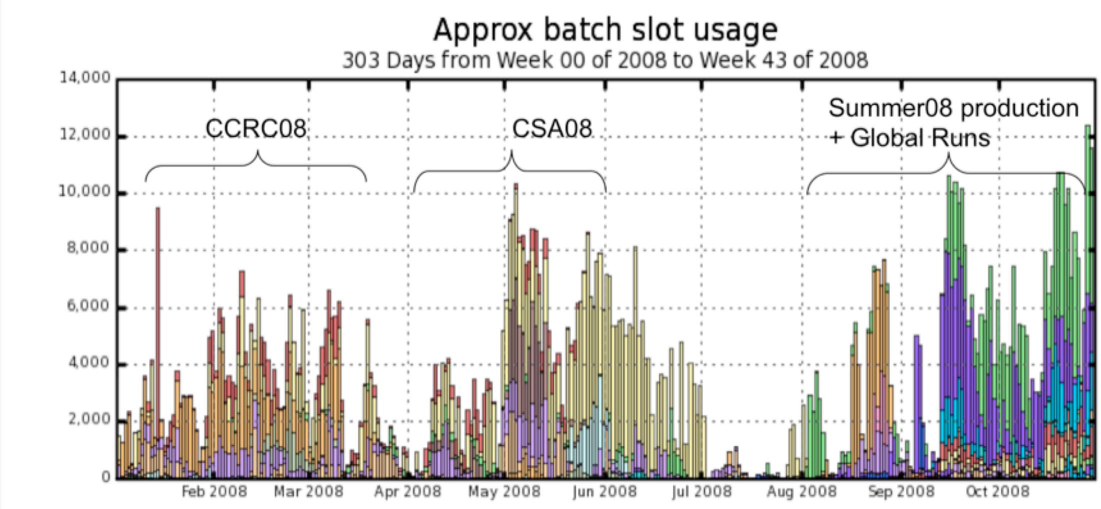

Figure 3: Approximate number of concurrent CPU slots occupied during 2008, each data series represents a different ProdAgent instance.

4.Future Plans

Further development is planned to ensure ProdAgent can meet the increased computing requirements as data taking progresses. Improvements are planned to increase resource use, scalability of individual instances and to ease development and operation.

4.1Request and Production Manager.

Currently distribution of work to a ProdAgent is a manual process - an operator is given a CMSSW config file that is then combined with details of the request (number of events, etc.) to create a workflow. This is then passed to the ProdAgent that creates, and submits, jobs based on this information. As there are multiple ProdAgent instances the resources available to each are restricted and the distribution of work to resources is not automatically adjusted. A way to optimize and distribute work to instances will be provided by a new system, ProdMgr. This service will collect work allocations and be contacted by ProdAgents requesting work.

This will be combined with another new service, Request Manger, which will be the entry point for physics requesters to the production system, where requests will be created and managed. ProdMgr will contact this service to obtain the requested work, in the form of a workflow, which will then be distributed to ProdAgents for execution. It is envisaged that once a request is accepted a small subset of the requested data will be produced with a high priority, this will allow the sample to be validated. Once the fast turn around subset has been accepted the remainder of the sample will be processed.

4.2Code re-organization

The ProdAgent tool is currently used for all Monte Carlo Production, data processing and within the Tier0. There is also a large amount of shared functionality with the CMS distributed analysis tool (CRAB [8]). Changes in common layers require changes to both projects. To improve this situation it has been decided to share a common layer of code between the 2 projects. This will be split into 2 distinct areas: an area for common library functionality (WMCore) and another for common component functionality (WMAgent).

The WMCore area will contain common code for functionality such as workflow and job definitions and interaction with remote services (both grid and CMS specific). These will be used by other CMS projects that need this functionality. The CRAB project has recently developed a server version that uses the prodAgent component architecture. To improve on this the common areas between the 2 two projects will be pushed down to the WMAgent level, this will include functionality such as the message service, component framework and any shared agent functionality.

4.3Performance improvements

The scalability requirements for ProdAgent are expected to increase in line with the data collected by CMS. Ensuring ProdAgent can meet this goal is expected to be a large area of work in the coming year. Work has already been carried out on areas that were observed to hamper operations. Interactions with some grid middleware, for example, have been optimized. Changes were made by the adoption of code initially developed within the CRAB project, see [8] for details.

Work is also underway to modify the way files are registered with DBS, currently the component responsible for file registration can become backlogged so it is being re-written to buffer information and make fewer remote calls. Some of this work will be carried out simultaneously with the move to WMCore and WMAgent, thus ensuring that the new code base is suitable for future operations.

5.Conclusion

The CMS production system has been shown to scale to the desired levels for the start of CMS data taking. Further work is required to ensure the system scales for the future, especially given an expected future reduction in available operations effort.

6. Acknowledgements

This work is partly supported by US Department of Energy grant DOE DE-FG02-06ER86271 and US National Science Foundation grant NSF PHY-0533280.

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and don't necessarily reflect the views of the Department of Energy or NSF.

References

[1] M. Della Negra, et al, *CMS Computing Technical Design Report,* CERN-LHCC-2005-023

[2] Rene Brun and Fons Rademakers, *ROOT - An Object Oriented Data Analysis Framework*, Proceedings AIHENP'96 Workshop, Lausanne, Sep. 1996, Nucl. Inst. & Meth. in Phys. Res. A 389 (1997) 81-86. See also http://root.cern.ch/.

[3] D. Evans et al, *The CMS Tier 0*, in proceedings of *XII Advanced Computing and Analysis Techniques in Physics Research*, Nov. 3-7, 2008,PoS(ACAT08)036.

[4] A. Afaq, A. Dolgert, Y. Guo, C. Jones, S. Kosyakov, V. Kuznetsov, L. Lueking, D. Riley, V. Sekhri, *The CMS Dataset Bookkeeping System*, Journal of Physics, Conferences Series (119) 072001, 2008

[5] R. Egeland, T. Wildish, S. Metson, *Data Transfer Infrastructure for CMS Data Taking*, in proceedings of *XII Advanced Computing and Analysis Techniques in Physics Research*, Nov. 3-7, 2008, PoS(ACAT08)033.

[6] MySQL, www.mysql.com

[7] I. Fisk, *Early Experience with the CMS Computing Model*, in proceedings of *XII Advanced Computing and Analysis Techniques in Physics Research*, Nov. 3-7, 2008, PoS(ACAT08)038

[8] G. Codispoti, *Distributed analysis in CMS using CRAB: the client-server architecture evolution and commissioning*, in proceedings of *XII Advanced Computing and Analysis Techniques in Physics Research*, Nov. 3-7, 2008, PoS(ACAT08)029.