CMS Monte Carlo production in the WLCG computing Grid

J M Hernández[1], P Kreuzer[2], A Mohapatra[3], N De Filippis[4], S De Weirdt[5], C Hof[2], S Wakefield[6], W Guan[7], A Khomitch[2], A Fanfani[8], D Evans[9], A Flossdorf[10], J Maes[5], P van Mulders[5], I Villella[5], A Pompili[4], S My[4], M Abbrescia[4], G Maggi[4], G Donvito[4], J Caballero[1], J A Sanches[7], C Kavka[11], F van Lingen[12], W Bacchi[8], G Codispoti[8], P Elmer[13], G Eulisse[14], C Lazaridis[3], S Kalini[2], S Sarkar[15], G Hammad[16]

[1] CIEMAT, Madrid, Spain
[2] RWTH, III Physikalisches Institut, Aachen, Germany
[3] University of Wisconsin, Madison, WI, USA
[4] Dipartimento Interateneo di Fisica di Bari and INFN Sezione di Bari, Bari, Italy
[5] Vrije Universiteit Brussel, Brussel, Belgium
[6] Imperial College, London, UK
[7] CERN, Geneva, Switzerland
[8] Università degli Studi di Bologna and INFN Sezione di Bologna, Bologna, Italy
[9] FNAL, Batavia, IL, USA
[10] DESY, Hamburg, Germany
[11] INFN Sezione di Trieste, Trieste, Italy
[12] California Institute of Technology, Pasadena, CA, USA
[13] Princeton University, Princeton, NJ, USA
[14] Northeastern University, Boston, MA, USA
[15] INFN Sezione di Pisa, Pisa, Italy
[16] Université Libre de Bruxelles, Bruxelles, Belgium

E-mail: `jose.hernandez@ciemat.es`

Abstract. Monte Carlo production in CMS has received a major boost in performance and scale since last CHEP conference. The production system has been re-engineered in order to incorporate the experience gained in running the previous system and to integrate production with the new CMS event data model, data management system and data processing framework. The system is interfaced to the two major computing Grids used by CMS, the LHC Computing Grid (LCG) and the Open Science Grid (OSG).

Operational experience and integration aspects of the new CMS Monte Carlo production system is presented together with an analysis of production statistics. The new system automatically handles job submission, resource monitoring, job queuing, job distribution according to the available resources, data merging, registration of data into the data bookkeeping, data location, data transfer and placement systems. Compared to the previous production system automation, reliability and performance have been considerably improved. A more efficient use of computing resources and a better handling of the inherent Grid unreliability have resulted in an increase of production scale by about an order of magnitude, capable of running in parallel at the order of ten thousand jobs and yielding more than two million events per day.

1. Introduction

An efficient and performant Monte Carlo (MC) production system is crucial for delivering the large data samples of fully simulated and reconstructed events required for detector performance studies and physics analyses in CMS. The Worldwide LHC Computing Grid (WLCG) makes available a large amount of distributed computing, storage and network resources for data processing. While WLCG provides the basic services and resources for distributed computing, reliability and stability of both global Grid services and of sites remain key issues, given the complexity of services and heterogeneity of resources in the WLCG Grid. A robust, scalable, automated and easy to maintain MC production system that can make an efficient use of the WLCG distributed resources is mandatory.

CMS has developed such a system during the past few years. Based on the experience gained in integrating and operating the first implementation in WLCG, the MC production system has been recently re-engineered. Production performance and scale have got a major boost thanks to the new production system, a strengthened production organization and an improved operational expertise. Integration aspects, operational experience and performance of the new production system will be presented in this paper.

2. ProdAgent production system

The former CMS MC production system on the Grid, McRunjob [1], suffered from the lack of the proper automation, monitoring and error handling necessary when processing in a distributed environment with inherent instability and unreliability. As a consequence, the production system was not efficient and robust enough, did not scale beyond running about thousand jobs in parallel and was very manpower-intensive. There was therefore the need for a new production system to automatically handle job preparation, submission, tracking, re-submission and data registration into the various data management databases (data bookkeeping, location and transfer systems). In addition, the new CMS event data model and processing framework made it possible to simplify the quite complex processing and publication workflow in McRunjob. In particular, the lack of support for multi-step processing in a single job and the need of metadata generation from the event data in order to make the data available for further processing or analysis were two sources of big overhead.

In 2006 the system for large scale MC production in CMS was re-engineered. McRunjob was replaced by ProdAgent [2]. The new system was designed aiming at automation, ease of maintenance, scalability, avoid single points of failure and support multiple Grid systems. In addition, ProdAgent integrated the production system with the new CMS event data model, data management system and data processing framework.

The implementation of a new production system was an opportunity to incorporate the very valuable operational experience gained with McRunjob in LCG [3]. It was found during the integration and operation of production with McRunjob that built-in robustness in all interactions with Grid and site services is a key element for efficiency and performance. In particular, I/O operations with the local storage system at sites are fault-prone. Jobs also often fail during any of the various stages of the Grid workload management system (WMS). Automatic job tracking and re-submission are very important. A global file replica catalogue for registering production data was found not to scale to CMS needs. Therefore, CMS moved to a data management system (DMS) where files are not tracked individually but are grouped into file blocks which are registered, replicated and analyzed together. It was also realized that dealing with small files is inadequate for transfer and storage. Therefore, CMS implemented in all data processing workflows a merge step where files are merged into larger files of few GBs which are those that eventually get registered in the data management databases. Given the level of instability in the services and infrastructure at the sites, a good support and prompt responsiveness from sites was found to be critical. Production eventually developed into a

job submission strategy based on white lists of sites with a proven record of stability, level of support and size. It was also realized that tools for infrastructure availability monitoring were very necessary. CMS consequently developed a set of tools for continuously watching site services and infrastructure via the submission of monitoring jobs. Proper job prioritization and share between different activities (production and analysis) was also found to be missing. Production jobs are now submitted with a voms proxy production extension so that they can be mapped at sites into a specific local production user that can be configured to have certain job priority and share.

2.1. Production framework and workflow

Figure 1 sketches CMS data and workflow management services.

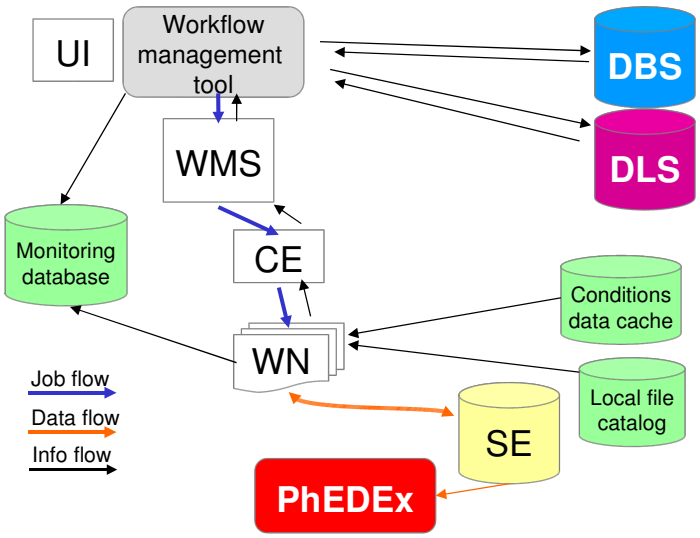

Figure 1. CMS data and workflow management services.

The workflow management tool, in this case ProdAgent, interacts with the DMS to discover data to process or to register produced data. The new CMS DMS is based on a set components that manage the large amounts of data produced, processed and analysed in the CMS distributed computing environment. The Dataset Bookkeeping Service (DBS [4]) describes the existing data (files, blocks, datasets). The Data Location Service (DLS [5]) tracks the location of the data blocks. Finally, local physical file names (PFN) and access protocols at the sites are provided to the jobs through a local Trivial File Catalogue (TFC), a set of logical to physical file name conversion rules based on a structured name space provided by the local storage system. ProdAgent submits production jobs to the Grid WMS. Jobs enter sites through a computing element (CE) which distributes them between worker nodes (WN). Jobs locally determine file PFNs via the local TFC and read conditions data via a local conditions data cache [6]. Data to process are read directly from the local storage system using the appropriate local posix-like I/O access protocol. Produced data are staged out into the local storage system. The CMS data transfer and placement system, PhEDEx [7], takes care of harvesting production files and transports the data to the analysis sites.

ProdAgent is built as a set of loosely coupled components that cooperate to carry out production workflows. Components are python daemons that communicate through a mysql database. Components use an asynchronous subscribe/publish model for communication. Their

states are persistently recorded in the database. Work is split into these atomic components that encapsulate specific functionalities.

Scaling is achieved by running in parallel any number of ProdAgent instances. Every instance makes use of a local ProdAgent mysql database for operation and monitoring of the components as well as a local DBS/DLS instance for data bookkeeping. Produced data are published into the data transfer system database and into the global DBS/DLS instance to make them available for transfer and to the collaboration for analysis (Figure 2).

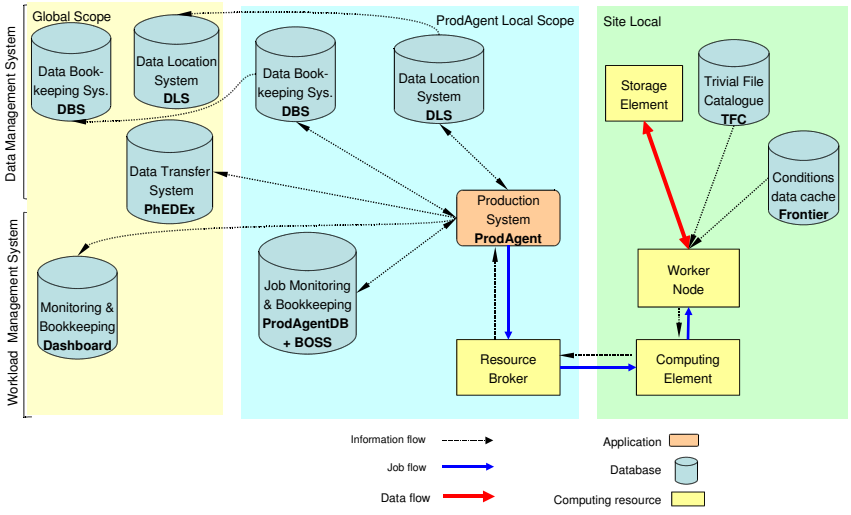

Figure 2. Production workflow with ProdAgent.

ProdAgent includes components for job creation, submission and tracking, error handling and job cleanup, data merging and publication into global DBS/DLS/data transfer system. The interface to the different Grid flavours is done via plug-in's for the job creation, submission and tracking components. It is fairly simple to add new systems and customization.

Each Job returns a detailed XML record of processing to the ProdAgent for accounting. It includes details of inputs, outputs, job information, any errors and diagnostics, etc. ProdAgent extends the job report with additional information like dataset, data tiers, site information, file checksums, etc. This report file is passed to components upon job completion for triggering data registration and data merge. The merge component triggers jobs based on the availability of data in the local DBS instance.

3. Production organization and performance

The aim of production operations is to deliver the largest possible amount of MC events to the collaboration by optimally using the available distributed computing resources. The most important ingredients to reach this goal are the automation of the production machinery and a high level of operational expertise, in particular for handling grid- or site-related issues.

The new CMS production machinery has been routinely used through various production rounds. It has brought a drastic boost in performance compared to the old McRunjob system, by increasing the number of production jobs from about 1k to about 20k per day. The current production scheme comprises a ProdRequest system, for physics groups to place their production requests and to create production workflows, a production coordinator, for organizing, approving and assigning workflows and several production teams that execute the assigned workflows by running a number of ProdAgent instances. An important aspect in this organization is to make sure that the software release, the configurations and the production system have been

properly and systematically validated, based on large sets of test samples ran prior to production assignments.

Production is typically run either as a single step (generation, simulation, digitization and reconstruction in the same job) or splitting the processing in two steps, generation+simulation, the most time-consuming part of the processing and not subject to change very often, and digitization+reconstruction, that can be redone whenever improved reconstruction code is available. The event processing time widely varies depending on the process being simulated. It ranges from less than one minute (all steps) for minimum bias inelastic events up to 700 seconds for multi-jet events. The typical event size is about 2 MB (0.5 MB simulation and 1.5 MB reconstruction information).

There are different levels of monitoring, bookkeeping and accounting in the production system. Production teams run a local ProdAgent monitor for each instance in order to watch production progression. Production statistics for all production instances are gathered in a central accounting database where the production manager can analyze production performance. Production jobs send reports to a central dashboard so that sites can look at production efficiency and discover potential problems. Information about produced data (number of files, events, blocks, datasets, locations, etc) is registered in the global DBS/DLS databases so that requestors can follow production progress and users can discover the data for analysis.

In the CMS computing model, MC production is conducted at the Tier-2 sites. Although Tier-1 sites are in charge of executing other workflows (organized data skimming and reprocessing), those centers are also used for MC production whenever no activity is scheduled and CPUs are idle. CMS has been routinely using about 35 Tier-2 centers for MC production (and up to 8 Tier-1 centers when available), hence up to about 10k of available batch slots (5.5k at the Tier-2's and 4.5k at the Tier-1's). These resources are divided geographically into several groups of sites, each of which is handled by a given production team (white list). The motivation for such a division is that a large fraction of production issues are related to site instabilities and hence require a good expertise by production operators in order to handle such issues in collaboration with the appropriate local site administrators. Up to 6 production teams have been simultaneously active during the major production rounds in year 2007, including 1 team for OSG sites and 5 teams for LCG sites. Various production rounds are listed in Table 1, including the number of produced events and the achieved average production rate. A given production team might be using several instances of ProdAgent, allowing parallel and faster submission of a large number of jobs. Table 1 also shows that the number of active teams and the number of ProdAgent instances have diminished with time whereas production rate has increased. This is due to the improving level of automation and performance of the production machinery.

Table 1. Production statistics for several production rounds.

Production period	Produced events	Production rate	Production teams / No of ProdAgent's
Winter07	90 M	35 M/month	6 / 10
Spring07	67 M	46 M/month	6 / 9
Summer07	145 M	64 M/month	5 / 7

The yield of events produced by site during the last part of the Spring07 and during Summer07 rounds is shown in Figure 3. The production rate is increasing with time as can be seen from the slope of the graph. For the period shown in the figure, a total of 175M events were produced, 2/3 of which were produced at Tier-2 sites and about 50% at OSG sites.

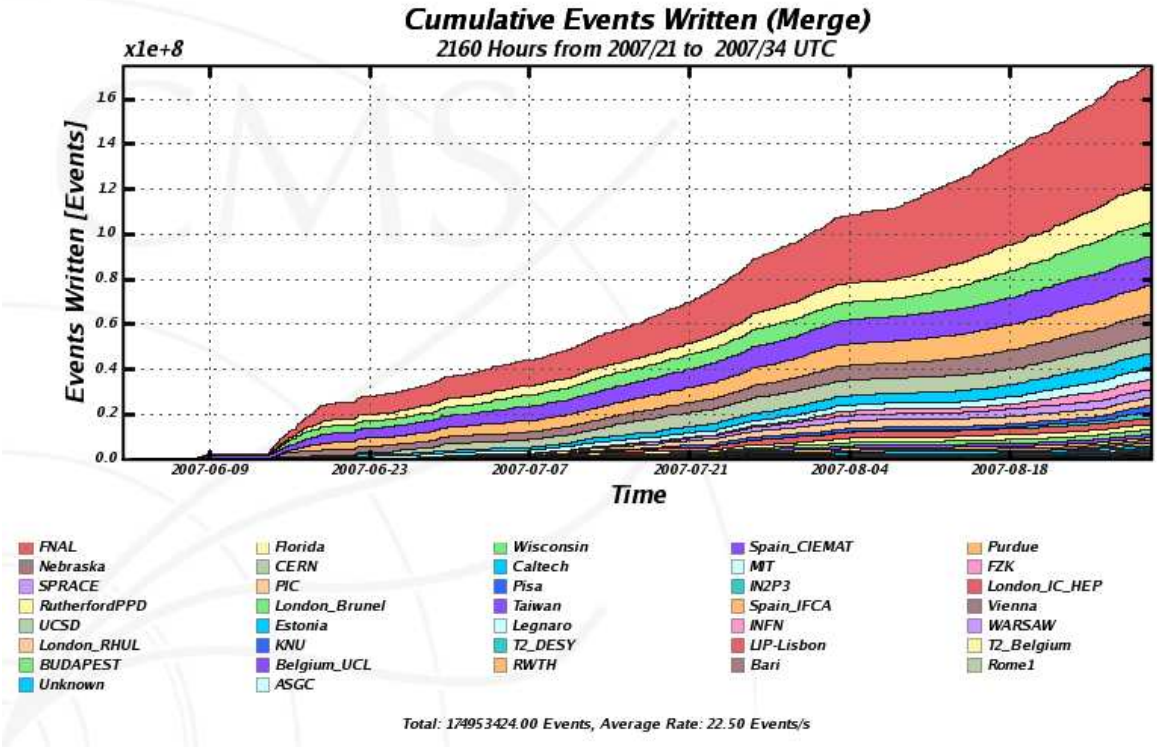

Figure 3. Accumulated number of events produced by site.

Figure 4 shows the success rate for production jobs for the various production sites as a function of time. It can be seen that more than 40 sites have been routinely used for production high high job efficiency. It should be noticed that failed jobs are automatically re-submitted by ProdAgent so that eventually most jobs succeed. The average job efficiency is around 75%, including application and Grid-related inefficiencies.

The number of production jobs completed daily classified by exit code is plotted as a function of time in Figure 5. In average about 20000 jobs are run every day.

About 80% of the failed jobs correspond to jobs aborted by the Grid. There are various reasons for aborted jobs: at submission time when for some reason the requirements of the jobs are met by no site (e.g. temporary problems with site publication in the Grid information system, experiment software release not installed, data location inconsistencies, temporary glitches in Grid services such as in resource brokers, computing elements, etc), at job running time when jobs are terminated abnormally (power failures in worker nodes, jobs killed by the local batch system when they exceed the allocated running time, failure in transporting the output sandbox to the resource broker via the local computing element, etc). Around 10% of the failures are accounted by data stage out problems when copying the job output file into the local storage element and 10% of the jobs fail due to application errors (failures accessing the input data, application crashes, access problems to the experiment software, etc).

In order to estimate the actual resource utilization level, i.e., the fraction of the available CPUs occupied continuously with production jobs, the total amount of CPU hours accumulated by production jobs every day divided by 24 hours is plotted in Figure 6 as a function of time. The different colors show the contribution from the various production teams. From June 2007, after production ramp-up, in average about 5000 slots have been filled 24/7 with production jobs. This occupancy corresponds to about 50% of the available resources pledged for MC production.

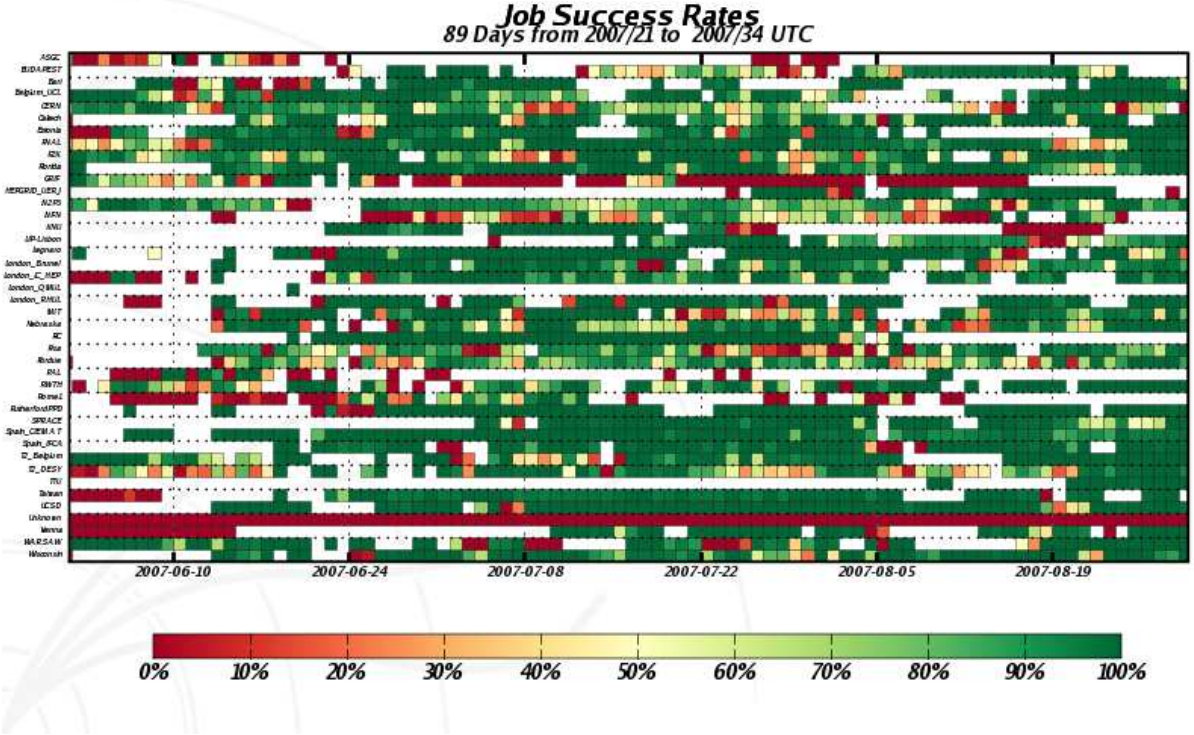

Figure 4. Job success rate for the various production sites.

In the most efficient production period during July07, the number of available resources was of 6300 batch-slots, while the achieved average was of 4600, hence 73% resource occupation (57% for Tier-2s only).

There are various reasons that explain the inability of fully efficiently using all available resources: (a) site-related issues (failing computing elements, problems in disk access or data stage out, etc), which need to be monitored and handled by production teams in close collaboration with site administrators; (b) production related issues, such as no automatic workflow management (manual job injection and release by operators) or inefficient workflow management (many sites and many small production requests).

The current production system is still subject to improvements in automation (see next section), and hence the resource occupation due to production issues is expected to increase, eventually leading to a nearly self-sustained system in terms of job submission and job tracking, in which the main role of production teams will be to monitor and handle grid-related or site-related issues.

4. New production components

There are new production components under final development, testing, integration or deployment that will enhance automation, robustness, performance and scale in the production system reducing at the same time the manpower required for operation.

Workflow management is being automated by means of two new components, the already deployed ProdRequest system that automates workflow creation by physics groups and operation managers, and the ProdManager component that supplies work (number of events to process, priority, etc) to the ProdAgents based on workflows entered in the ProdRequest system.

A much better use of the available resources will be attained by means of the new

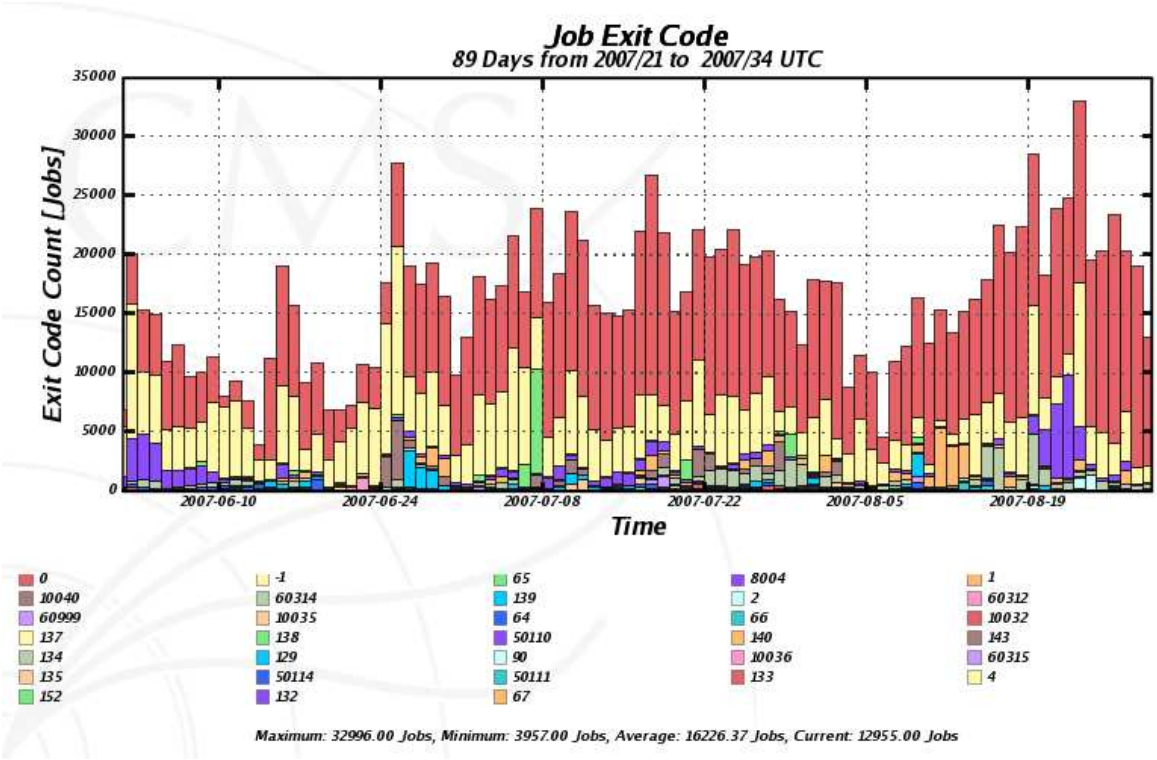

Figure 5. Number of production jobs completed daily classified by job exit code.

ResourceMonitor component. This component will watch for CPU resources available for production (polling the Grid information system or local ProdAgent job tracking monitors) and will automatically release jobs previously injected by ProdManager and queued at ProdAgent level.

Multi-step processing (concurrently processing of several workflows where the input data of a workflow is the output of the other) will be automated by introducing a new sensor component which polls the data bookkeeping service for available merged input data to trigger their processing.

Concerning scale issues, the number of jobs a single ProdAgent instance can handle will be largely increased by introducing bulk operations for job creation, submission and tracking.

In order to avoid that production jobs are killed by the local batch system when they exhaust the maximum time allocated to the job, and to avoid hanging jobs (for example waiting forever for the input data) that block batch slots for the duration of the slot not doing any processing, it is planned to move from event-based jobs to time-based jobs. Jobs will not be prepared to process a given number of events but to last for certain amount of time, adjusted to the queue length at sites.

The global monitoring and accounting system for production is being extended by means of a new monitoring component in ProdAgent. This component of every ProdAgent instance will periodically send to a global server in a reliable way statistics of completed jobs that can be later analyzed providing a valuable information for optimization of the production system. This extension will complement the current global monitoring system based on individual job reports from the worker nodes sent to a global dashboard via a UDP-based messaging system.

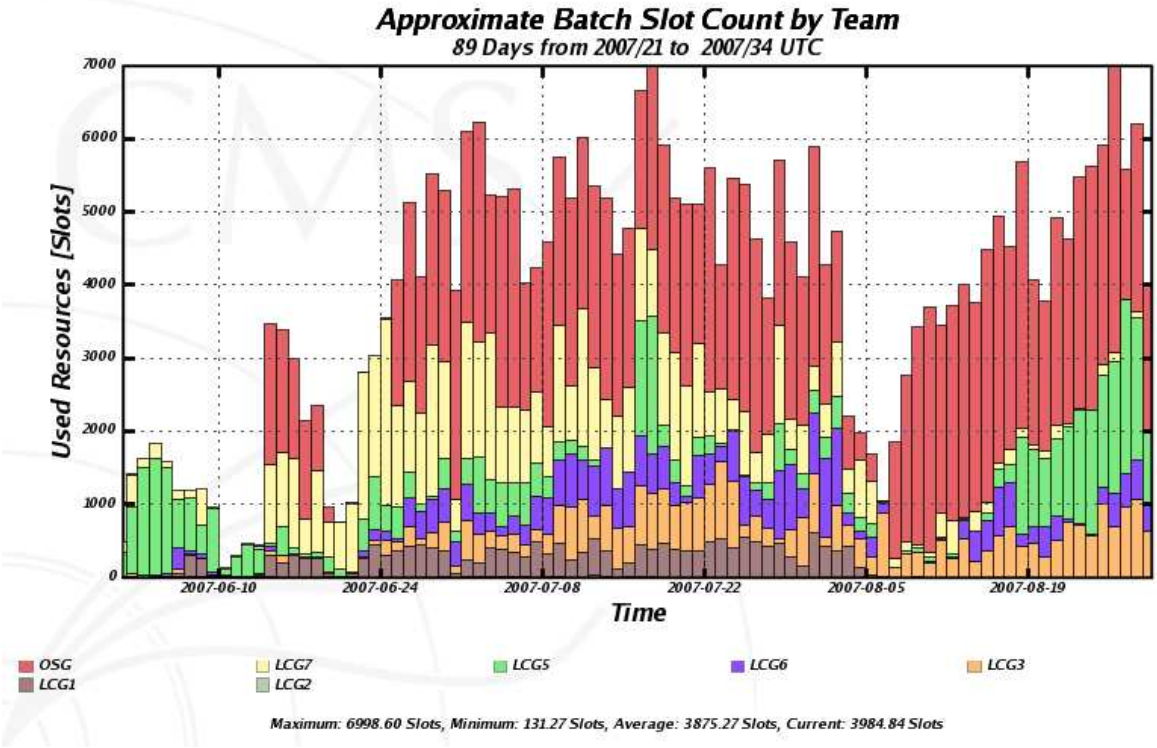

Figure 6. Approximate number of slots occupied continuously with production jobs.

5. Summary and conclusions

Monte Carlo production on the Grid in CMS has experimented a major boost in performance and scale since the last CHEP conference. The production system has been re-factored in order to incorporate the experience gained in integrating and running the former production system on the Grid and to adapt the system to the new CMS event data model, processing framework and data management system.

The new production system has dramatically improved automation, scalability, robustness and efficiency. Through a set of loosely coupled components production workflows are carried out in an automated fashion. Data processing and management is optimized by running multiple processing steps in a single job and by incorporating a data merging step in the production workflow. The system is fully coupled to the various data management system components. Each production instance runs its own local data and job tracking systems. Produced data are then promoted to the global data management databases, including the data transfer system. The new system brings scalability through the possibility of running any number of production instances in parallel. Each instance can currently handle few thousand concurrent jobs. A production scale of more than 20000 jobs per day, in average about 5000 concurrent jobs running continuously, has been reached by running several production instances in parallel. A production yield of 2 million events per day, with a job efficiency of about 75%, is routinely reached.

Production organization has also been strengthened. Bringing regularly together requestors, consumers, producers, developers and sites has also contributed to the increase in scale.

Performance in the production system is greatly affected by the unreliability and instability of global Grid services and sites (local storage and batch systems) or even unavailability for extended periods. Responsiveness of Grid and site administrators is crucial for an efficient production.

Running production in LCG is still a manpower-consuming task. Further automation is being implemented in the production system. New components will take care of continuous job injection based on available computing resources or available data to be processed. Workflow management will be further automated coupling the system with a production manager component which in turn gets production work to be done from a production request system. Bulk operations will allow to increase the scale of jobs a single production instance can handle. Time-based jobs will allow to use more efficiently the computing resources. An extended global monitoring and accounting system will allow to analyze production performance and usage of resources for further optimization.

References

[1] Caballero J, Garcia-Abia P and Hernández J M 2006 CMS Monte Carlo Production in the LHC Computing Grid, *Proc. Int. Conf. on Computing in High Energy and Nuclear Physics (Mumbai)* vol 2 (Macmillan India ltd.) p 974.

Garcia-Abia P and Hernández J M 2005 Implementation of Monte Carlo Production in LCG, *CMS-NOTE-2005/019.*

[2] Evans D et al 2007 CMS MC Production System Development and Design, *these proceedings.*

[3] Caballero J, Garcia-Abia P and Hernández J M 2007 Integration and operational experience in CMS Monte Carlo production in LCG, *CMS-NOTE-2007/016.*

[4] Lueking L et al 2007 The CMS Dataset Bookkeeping Service, *these proceedings.*

[5] Fanfani A et al 2006 Distributed Data Management in CMS *Proc. Int. Conf. on Computing in High Energy and Nuclear Physics (Mumbai)* vol 2 (Macmillan India ltd.) p 1006.

[6] Lueking L et al 2007 CMS Conditions Data Access using FroNTier, *these proceedings.*

[7] Rehn J, Barrass T, Bonacorsi D, Hernandez J M, Semeniouk I, Tuura L and Wu Y 2006 PhEDEx high-throughput data transfer management system, *Proc. Int. Conf. on Computing in High Energy and Nuclear Physics (Mumbai)* vol 2 (Macmillan India ltd.) p 1027.

Barrass T, Bonacorsi D, Hernandez J M, Rehn J, Tuura L and Wu Y 2006 Techniques in high-throughput, reliable transfer systems: break-down of PhEDEx design *Proc. Int. Conf. on Computing in High Energy and Nuclear Physics (Mumbai)* vol 2 (Macmillan India ltd.) p 1030.

Tuura L et al 2007 Scaling CMS data transfer system for LHC start-up, *these proceedings.*