J Grid Computing
DOI 10.1007/s10723-010-9152-1

Distributed Analysis in CMS

Alessandra Fanfani · Anzar Afaq · Jose Afonso Sanches · Julia Andreeva · Giusepppe Bagliesi · Lothar Bauerdick · Stefano Belforte · Patricia Bittencourt Sampaio · Ken Bloom · Barry Blumenfeld · Daniele Bonacorsi · Chris Brew · Marco Calloni · Daniele Cesini · Mattia Cinquilli · Giuseppe Codispoti · Jorgen D'Hondt · Liang Dong · Danilo Dongiovanni · Giacinto Donvito · David Dykstra · Erik Edelmann · Ricky Egeland · Peter Elmer · Giulio Eulisse · Dave Evans · Federica Fanzago · Fabio Farina · Derek Feichtinger · Ian Fisk · Josep Flix · Claudio Grandi · Yuyi Guo · Kalle Happonen · José M. Hernàndez · Chih-Hao Huang · Kejing Kang · Edward Karavakis · Matthias Kasemann · Carlos Kavka · Akram Khan · Bockjoo Kim · Jukka Klem · Jesper Koivumäki · Thomas Kress · Peter Kreuzer · Tibor Kurca · Valentin Kuznetsov · Stefano Lacaprara · Kati Lassila-Perini · James Letts · Tomas Lindén · Lee Lueking · Joris Maes · Nicolò Magini · Gerhild Maier · Patricia Mcbride · Simon Metson · Vincenzo Miccio · Sanjay Padhi · Haifeng Pi · Hassen Riahi · Daniel Riley · Paul Rossman · Pablo Saiz · Andrea Sartirana · Andrea Sciabà · Vijay Sekhri · Daniele Spiga · Lassi Tuura · Eric Vaandering · Lukas Vanelderen · Petra Van Mulders · Aresh Vedaee · Ilaria Villella · Eric Wicklund · Tony Wildish · Christoph Wissing · Frank Würthwein

Received: 23 August 2009 / Accepted: 2 March 2010

Abstract The CMS experiment expects to manage several Pbytes of data each year during the LHC programme, distributing them over many computing sites around the world and enabling data access at those centers for analysis. CMS has identified the distributed sites as the primary

A. Fanfani (✉) · D. Bonacorsi · G. Codispoti · C. Grandi
INFN and University of Bologna, viale Berti Pichat 6/2, 40127 Bologna, Italy
e-mail: fanfani@bo.infn.it

A. Afaq · L. Bauerdick · D. Dykstra · D. Evans · I. Fisk · Y. Guo · C.-H. Huang · L. Lueking · P. Mcbride · P. Rossman · V. Sekhri · E. Vaandering · E. Wicklund
Fermilab, Batavia, IL, USA

J. Afonso Sanches · P. Bittencourt Sampaio
University of Rio De Janeiro UERJ, Rio De Janeiro, Brazil

J. Andreeva · M. Calloni · N. Magini · V. Miccio · P. Saiz · A. Sciabà · D. Spiga
CERN, Geneva, Switzerland

G. Bagliesi
Pisa INFN, Pisa, Italy

S. Belforte · C. Kavka
Trieste INFN, Trieste, Italy

K. Bloom
University of Nebraska, Lincoln, NE, USA

B. Blumenfeld
Johns Hopkins University, Baltimore, MD, USA

C. Brew
Rutherford Appleton Laboratory, Didcot, UK

D. Cesini · D. Dongiovanni · N. Magini · V. Miccio
INFN-CNAF, Bologna, Italy

M. Cinquilli · H. Riahi · A. Vedaee
Perugia INFN, Perugia, Italy

Published online: 20 March 2010

location for physics analysis to support a wide community with thousands potential users. This represents an unprecedented experimental challenge in terms of the scale of distributed computing resources and number of user. An overview of the computing architecture, the software tools and the distributed infrastructure is reported. Summaries of the experience in establishing efficient and scalable operations to get prepared for CMS distributed analysis are presented, followed by the user experience in their current analysis activities.

Keywords LHC · CMS · Distributed analysis · Grid

1 Introduction

The Compact Muon Solenoid (CMS) [1] is a general-purpose detector built to collect data at the Large Hadron Collider (LHC), located at CERN (Geneva, Switzerland). The beams will collide at intervals of 25 ns and CMS will record only the collisions that pass a set of online trigger decisions with an expected rate around 300 Hz and an average event size of 1–2 MB. A nominal-year worth of data taking corresponds to about 2–6 PB of storage prior to any processing. Data will have to be accessed for reprocessing and analysis by a large experimental community with more

J. D'Hondt · J. Maes · P. Van Mulders · I. Villella
Brussel University, Brussel, Belgium

L. Dong
Institute of High Energy Physics, Chinese Academy of Sciences Academia Sinica, Beijing, China

G. Donvito
INFN and University of Bari, Bari, Italy

E. Edelmann · K. Happonen · J. Klem · J. Koivumäki · K. Lassila-Perini · T. Lindén
Helsinki Institute of Physics, Helsinki, Finland

R. Egeland
University of Minnesota, Twin Cities, MN, USA

P. Elmer · T. Wildish
Princeton University, Princeton, NJ, USA

G. Eulisse · L. Tuura
University of Northeastern, Boston, MA, USA

F. Fanzago
Padova INFN, Padova, Italy

F. Farina
Milano Bicocca INFN, Milan, Italy

D. Feichtinger
Paul Scherrer Institut (PSI), Villigen, Switzerland

J. Flix · J. M. Hernàndez
CIEMAT, Madrid, Spain

J. Flix
PIC, Barcelona, Spain

K. Kang
Peking University, Peking, China

E. Karavakis · A. Khan
Brunel University, London, UK

M. Kasemann · C. Wissing
DESY, Hamburg, Germany

B. Kim
University of Florida, Gainesville, FL, USA

T. Kress · P. Kreuzer
RWTH, Aachen, Germany

T. Kurca
Institut de Physique Nucleaire de Lyon, Villeurbanne Cedex, France

V. Kuznetsov · D. Riley
Cornell University, Ithaca, NY, USA

S. Lacaprara
Legnaro INFN, Legnaro, Italy

J. Letts · S. Padhi · H. Pi · F. Würthwein
University of California San Diego, La Jolla, CA, USA

G. Maier
University of Linz, Linz, Austria

than 3,000 collaborators (CMS Collaboration, http://cms.web.cern.ch/cms/Collaboration/index.html) distributed worldwide across 40 countries. This imposes an unprecedented computing challenge for data management and processing.

2 The CMS Computing Model

The CMS distributed computing and analysis model [2, 3] is designed to serve, process and archive the large number of events which will be generated when the CMS detector starts taking data. The computing resources are geographically distributed, interconnected via high-throughput networks and accessed by means of Grid techniques. The choice of a distributed system allows delegation of responsibilities to local CMS communities, access to additional funding channels and ensures load balancing of the available resources while replicating the interesting data in different sites.

A multi-Tier hierarchical distributed model is adopted in CMS with specific functionality at different levels.

Tier-0 The Tier-0 centre at CERN accepts data from the CMS online system, archives the data, performs prompt first pass reconstruction. Reconstructed data at the Tier-0 together with the corresponding raw data are distributed to Tier-1s over the Optical Private Network that is the backbone network specifically built for LHC to interconnect CERN and the Tier-1s. In addition to the Tier-0 centre, CERN hosts the CMS Analysis Facility (CAF) that is focused on latency-critical detector, trigger and calibration activities. Roughly 20% of the computing capacity is located at CERN, while the remainder is distributed.

S. Metson
Bristol University, Bristol, UK

A. Sartirana
Ecole Polytechnique, Paris, France

L. Vanelderen
University of Gent, Gent, Belgium

Tier-1 Each Tier-1 centre assures the custodial storage of a fraction of the raw data produced by the CMS detector and of the simulated data produced at the connected Tier-2 centres. Tier-1 centres provide computing resources for their further re-processing (re-reconstruction, skimming, etc.) and for high priority analysis. They control the data transfer to the Tier-2 centres and among them for analysis. There are 7 Tier-1 centres located in France, Germany, Italy, Spain, Taiwan, UK and USA.

Tier-2 Tier-2 centers, about 50 sites around the world [4, 5], provide capacity for user data analysis and for production of simulated data. In the CMS data transfer topology, transfers to Tier-2 can occur from any Tier-1. A significant effort is required in commissioning all needed transfer links, as described in Section 4.2.2, as well as improving site availability and readiness, as described in Section 4.2.1.

3 Framework for CMS Distributed Analysis

The CMS analysis model foresees activities driven by data location. Data are distributed over many computing centers according to CMS data placement policies. Processing takes place in the sites where data are located. In order to enable distributed analysis, a set of Workload and Data Management tools have been developed, building CMS-specific services on top of existing Grid services.

3.1 Data Management

The CMS Data Management System provides the basic infrastructure and tools necessary to manage the large amounts of data produced, processed and analysed in a distributed computing environment. In order to simplify bulk data management, files are grouped together into file-blocks of a convenient size for data transfer. File-blocks are in turn grouped in datasets whose content is driven by physics. The file-block is the unit of data location and replication. The tracking of data

location is file-block based and it provides the name of sites hosting the data, not the physical location of constituent files at the sites nor the composition of file-blocks. The file-block contains files that can be processed and analyzed together. The packaging of events into files is done so that the average file size is kept reasonably large (e.g. at least 1 GB), in order to avoid scaling issues with storage and tape systems and optimize data transfer. This is achieved by merging small output files produced by individual jobs into fewer larger files.

The CMS Data Management System is made of a set of loosely coupled components as described in the following sections.

3.1.1 DBS

The Dataset Bookkeeping Service (DBS) [6] provides the means to describe, discover and use CMS event data. It catalogs CMS specific data definitions such as run number, the algorithms and configurations used to process the data together with the information regarding the processing parentage of the data it describes. The DBS stores information about CMS data in a queryable format. The supported queries allow discovery of available data and the way they are organized logically in term of packaging units like files and file-blocks. The information available from queries to DBS are site independent.

The DBS is used for data discovery and job configuration by the production and analysis systems through a DBS API. Users can discover which data exist using either a Web browser or a command line interface. The DBS is usable in multiple "scopes":

- A Global scope DBS is a single instance describing data CMS-wide;
- Many local-scopes DBS's are established to describe data produced by MonteCarlo production, Physics groups or individuals. Data produced in local-scope may be migrated to global-scope as needed.

The DBS system is a multi-tier web application with a modular design. This makes it easily adaptable to multiple database technologies. The supported types of database (Oracle, MySQL and SQLite) enable the DBS deployment in a range of environments from general CMS at large installations to specific personal installations. XML is used as the format of the HTTP payload exchanged with the client.

The Global DBS is hosted at CERN and its database engine is the CERN Oracle RAC (Real Application Cluster) server for CMS. Some local-scope DBS instances that catalog data from Physics groups are also hosted at CERN. There are also DBS instances installed at other sites for private use.

3.1.2 Local Data Catalogue

A CMS application only knows about logical files and relies on a local catalogue service to have access to the physical files. Each CMS site has a Trivial File Catalogue made of simple rules to build site-specific physical paths starting from logical file names and access protocols.

3.1.3 Conditions Data

The data describing the alignment and calibration of the detector are known as "conditions data". Since the same conditions data need to be accessed by many processing jobs worldwide CMS uses a caching system called FroNTier [7]. FroNTier translates database queries into HTTP, looks up the results in a central database at CERN, and caches the results in a industry-standard HTTP proxy/caching server called Squid [7]. Squid servers are deployed at each site. Conditions data is read by the applications from these Squid servers.

3.1.4 PhEDEx

The CMS data placement and transfer systems are implemented by PhEDEx [8, 9]. The data placement system provides an interface to define, execute and monitor administrative decisions of data movement such as where experimental data have to be located, which copies are custodial. Data are distributed according to available resources

and physics interests at sites as determined by CMS Data Operations, physics analysis groups, and/or the members of the "local" CMS community served by the site.

In PhEDEx, distinct storage areas (Grid sites or disk/tape areas within a site) are represented by a "node". Links between the "nodes" define the transfer topology. The transfer workflow begins when a user makes a transfer request of some data to some "node" via the web page, which is then approved by that "node"'s Data Manager. In the request, the user only specifies the destination "node", and the optimal source "node" is determined from among the available file replicas. To do this, PhEDEx uses Dijkstra's algorithm to calculate the path of least cost, where cost of transfer for each link is determined by the recent transfer rate and the size of the current queue over that link. Using this method, PhEDEx balances exports among multiple sources when the same data is requested to multiple "node"s. Additionally, it is fault-tolerant when links fail to perform and another source replica is available.

From a design standpoint, PhEDEx is based on software "agents" storing their state and communicating via a central "blackboard" database hosted in a CERN Oracle RAC installation. A set of service "agents" run centrally at CERN while each site in general runs only the "agents" that interact with the storage at the site. The usual method of transfer execution is to submit a job to the gLite File Transfer System (FTS), which is managed by the site download "agent" using the FTS backend. Download "agent" backends for other transfer methods are available, making PhEDEx technology independent of the underlying transfer mechanism. The PhEDEx web site offers major workflow management tools, including the request creation and approval interfaces, and allows users and site administrators to monitor current and historical transfer conditions. File deletion and on-demand consistency checking are also provided by "agents" running at the site receiving work queues from the central database. A web data service provides machine-readable XML or JSON data from the database, which is used to integrate PhEDEx with other CMS computing components. For instance, PhEDEx keeps track of data location in the distributed computing system and the analysis system relies on the data service to obtain the locations of the data when submitting jobs.

Using PhEDEx, CMS has transferred over 87 PB of data since the project began in 2004. Figure 1 shows the average daily transfer volume per month of data managed by PhEDEx

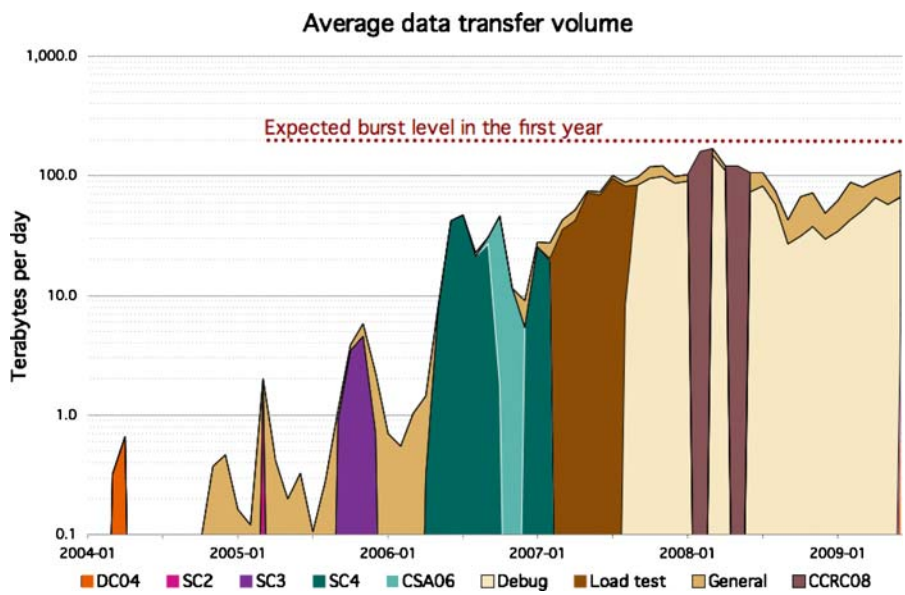

Fig. 1 Average daily transfer volume per month using PhEDEx since the project began

from mid-2004 until July 2009. Various challenge periods are highlighted, and in particular the SC4, CSA'06 [10–12], and LoadTest/Debug periods which resulted in large increases in the average transfer volume. "General" transfers include transferring of Monte Carlo or cosmics raw data for physics analysis. "Debug" and "LoadTest" transfers are of randomly generated data for the purpose of debugging and commissioning the transfer links at the fabric level. In the first 6 months of 2009, PhEDEx has sustained on average transfer volumes of over 80 TB per day, where roughly 40% of the traffic has been "General" production/analysis WAN traffic and the remaining 60% "Debug" commissioning traffic. Expected burst levels on any link during LHC data taking are at 200 TB per day. PhEDEx already demonstrated to be able to cope with these levels during the Common Computing Readiness Challenge (CCRC'08) challenge [13], in February 2008 and June 2008 (phase 1 and 2 respectively).

3.2 Workload Management

The role of the CMS Workload Management system includes the interpretation of user processing requests, the creation of the jobs which will process the data, the submission of the jobs to local or distributed systems, the monitoring of the jobs and the retrieval of their outputs. The Production Agent (ProdAgent) [14] is a tool optimized to perform these operations in a controlled environment i.e. at the Tier-0 and at the Tier-1 centres. The CMS Remote Analysis Builder (CRAB) is optimized for user analysis, as described in the next section.

3.2.1 CRAB

The CMS Remote Analysis Builder (CRAB) [15] has been developed as a user-friendly interface to handle data analysis in a local or distributed environment, hiding the complexity of interactions with the Grid and CMS services. It allows the user to run over large distributed data samples with the same analysis code he has developed locally in a small scale test.

The functionalities that CRAB provides, as schematically illustrated in Fig. 2, are:

- *Data discovery and location:* Facilitate queries of the experiment data catalogues (DBS and PhEDEx) to find what data exist and where they can be accessed.
- *Job preparation:* Pack local user code and the environment to be sent to remote sites where the CMS software is pre-installed as described in Section 4.1.3.
- *Job splitting:* Decide how to configure each job to access a subset of files in the dataset to effectively use the Tier-2s resources.
- *Job submission:* Submit to Grid sites hosting the required data.
- *Job monitoring:* Monitor the status of the submitted jobs by querying Grid services.
- *Handling Output data:* Copy the produced output to a remote Tier-2 the user is associated with or return it to the user for small files (few

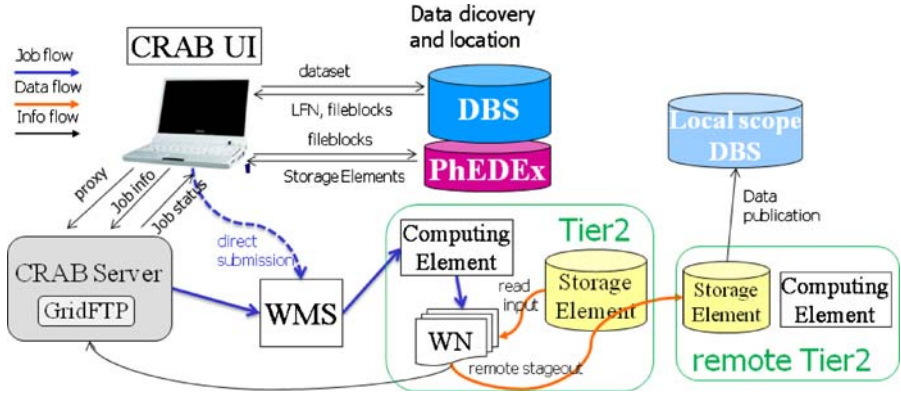

Fig. 2 CRAB workflow. WMS can refer either to gLite-WMS or glidein-WMS

MB). Publish the produced data with their description and provenance into a local DBS so that the data can be used in further analysis and shared with colleagues.

CRAB is coded in Python. The interface to the Grid middlewares and local batch systems is provided by a Python library named BOSSLite [16]. It relies on a database to track and log information about the user requested task into an entity-relation schema. An abstract interface is used to access the database through safe sessions and connection pools to grant safe operation in a multiprocessing/multi threaded environment. The current implementation supports MySQL and SQLite databases. Standard operations such as job submission, tracking, cancellation and output retrieval are also performed via a generic abstract interface. Scheduler-specific (batch system or Grid) interfaces are implemented as plug-ins, loaded at run-time. Presently, plug-ins are implemented for the Grid middleware EGEE (gLite-WMS) [17], OSG [18] both direct Condor-G submission or a pilot based job submission system (glidein-WMS [19]), and ARC (NorduGrid) [20].In addition plug-ins for batch systems such as LSF and SGE are provided.

The interaction with the Grid can be either direct with a thin CRAB client or using an intermediate CRAB Analysis Server [15] (see Fig. 2). The CRAB Analysis Server automates the user analysis workflow with resubmissions, error handling, output retrieval thus leaving to the user just the preparation of the configuration file and notifying him of the output availability. In addition it has the capability of implementing advanced analysis use cases. The CRAB Analysis Server is made of a set of independent components communicating asynchronously through a shared messaging service and cooperating to carry out the analysis workflow. The communication between client and server is implemented using the gSOAP framework and Grid credentials of users are delegated to server. The CRAB Analysis Server is coupled with an external GridFTP server that stores the user input and output data, allowing implementation of CMS policies on sandbox sizes, bypassing for instance the gLite-WMS limits.

3.3 Job Monitoring

Monitoring tools are critical to the success of the highly distributed analysis scenario in CMS. Good monitoring has allowed CMS to evolve tool development and operational structure from vision and anecdote driven to a fact based approach. A few main points drove the development of monitoring tools:

- no reliance on local site monitoring.
- a single high level view of the usage from which to drill down to single job level
- keep the system lean and flexible: even if a few jobs are not properly reported
- record enough information about failures so that plans and actions are set based on quantitative facts and that the effectiveness of solutions can be measured
- detect overall usage patterns to guide management in making choices and plans about how and where to steer user activities and how to plan for the future
- do not try to define a priori all the relevant metrics

Job monitoring is built around the idea of instrumented application: CMS jobs and tools send messages to a central collector. Only jobs which use the instrumented submission framework can be monitored in this way, a small penalty in the CMS case where almost all user jobs are submitted using the CRAB tool. The implementation is based on a database running on an Oracle server, a set of information collectors feeding from various sources, and a few web interfaces (views) providing access with different levels of detail, aggregation and flexibility, customized to typical use cases. It is possible to cross link and navigate from one view to another providing both extreme flexibility and fast access to desired information. This set of tools is called “the CMS Dashboard” [21].

3.3.1 History View

The aim here is to present time history of relevant metrics to highlight overall patterns and trends. The list of viewable metrics is predefined, and

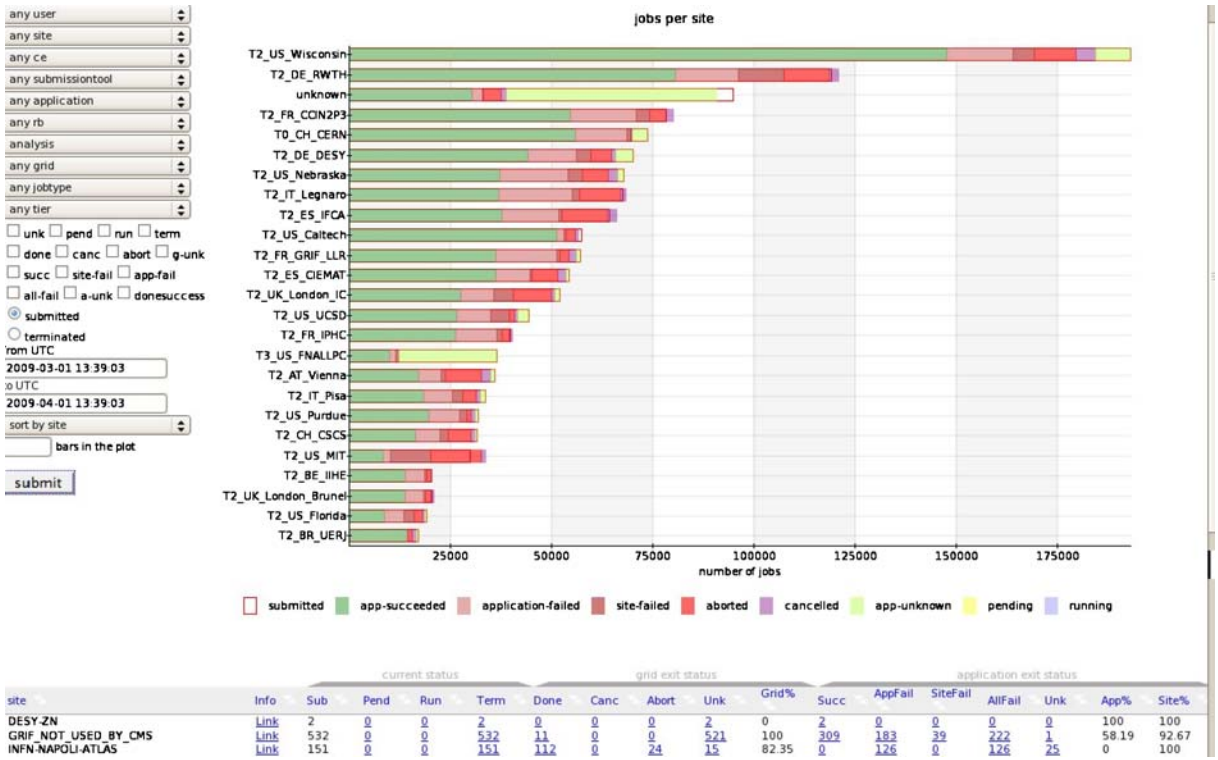

Fig. 3 Dashboard interactive interface

the interface uses aggregated tables in the DB to provide efficient access to old information with a limited amount of detail. Data can be viewed divided according to the used site(s) or job type or job completion status, among others.

3.3.2 Task Monitoring for Analysis User

A user-centric view where the user is initially presented with the list of tasks he has submitted in the last two days, both in tabular and graphical format.

The user can then expand one selected task to get a graphical overall summary of execution/completion progress and access details of each job.

3.3.3 Interactive Interface

This was the first view to be developed, based on vision more than experience, therefore emphasis was put on flexibility. It is a job-centric view aimed at understanding and debugging what happens "now". The entry point is the number of jobs submitted or terminated in a chosen time period (see Fig. 3).

The interactive interface allows to drill down expanding the set of jobs according to various relevant properties (execution site, Grid gateway, submitter user, completions status, Grid workload management host, activity type, used dataset etc.), until all details stored in the database about a chosen (set of) job(s) can be accessed. The interface reports success/failure according to Grid/site/application problem, and information on used wall-clock time and cpu time of jobs.

4 Operation of CMS Distributed Analysis

4.1 Distributed Infrastructure

The tools described above operate on a distributed infrastructure of sites offering uniform access via standard Grid services. Moreover CMS maintains software libraries at each site to get a

uniform execution environment. A small database is used to keep track of information specific to each site and relevant for CMS operations.

4.1.1 Grid Services

The access to CMS resources is controlled by Grid services provided by the WLCG project. Authorization is based on X.509 certificates extended with VOMS attributes that certify the membership of users in the CMS Virtual Organization and possibly to CMS groups and the roles they have. VOMS servers are provided and maintained by the WLCG project [22]. Data are stored on systems exposing a SRM interface (Storage Elements). The specific implementation of the storage system is left to the sites (e.g. dCache, DPM, Castor, StoRM, etc...). The storage solution at CMS Tier-1 sites also foresees a tape-based backend to fulfill the custodial responsibility as from the Computing model, while CMS Tier-2 site are disk-based.

The access to computing resources (Computing Elements) is based on the Globus gatekeeper (on EGEE, currently via its LCG-CE implementation) or on the ARC-CE on NorduGrid. The CMS workload management system (CRAB and ProdAgent) may use, via the BossLite layer, several high level workload management tools provided by the different infrastructures (see Section 3.2.1).

All computing and storage resources publish information on their characteristics and state to an Information System based on the BDII provided by WLCG. The information is used both for monitoring and for resource selection.

The CMS data transfer system is implemented by PhEDEx which in turn depends on the gLite File Transfer System (FTS). WLCG maintains a number of FTS servers (at Tier-0 and Tier-1 sites) which provide the transfer channels on which the transfers take place.

4.1.2 SiteDB

SiteDB [23] is a catalogue of all CMS computing Tiers. It records the CMS resources at the site, the resources pledged for the future, and keeps track of CMS personnel at each site, including the roles and duties they fulfill in the collaboration. CMS has designed and built SiteDB because CMS relies on close contact with the sites it uses. The distributed nature of CMS computing makes it very useful to track the people's responsibilities, and to contact them on the basis of their role. Some site roles are technical (for instance running PhEDEx), while others are related to CMS computing policy (e.g. the site's Data Manager).

4.1.3 Software Installation

Distributed analysis relies on the experiment software being pre-installed on the remote sites for each release. The CMS software is packaged using the RedHat Package Manager (RPM), which is used by some major Linux distributions. Dependencies between packages are resolved with the help of the apt-get tool, which is also widely used in the Linux community. The whole CMS software suite is installed on a filesystem shared among all worker nodes of a site.

During the first setup of CMS software ("bootstrap") the underlying operating system is checked for all required packages. These are imported into a meta-RPM, which is installed in the CMS software area. From that point on all CMS installations are independent of the underlying operating system. In addition to the actual CMS software (CMSSW) releases some "external" packages are installed, e.g. clients for databases, various ROOT versions and GEANT4.

The installations themselves are performed by high priority Grid jobs. Those jobs are sent using a dedicated VOMS role in order to gain write access to the CMS software area. Once a release is installed, a tag is set on the site which publishes the availability of the release. The tags are used by the workload management systems to submit jobs to sites which provide the requested CMS release. Typically, there are about 10 production releases installed at a time at all sites. This requires roughly 50 GB of disk space.

All software deployment and removal activities are handled centrally using two different instances, one for OSG based sites and one for gLite and ARC based sites. Including Tier-1, Tier-2 and Tier-3 sites about 80 sites are served routinely.

4.2 Infrastructure Readiness

Operation of the CMS distributed infrastructure requires a stable and reliable behaviour of the services. CMS has established a procedure routinely operated by the Facilities Operations team to extensively test all relevant aspects of sites supporting CMS [4, 5], such as the ability to efficiently use their network to transfer data, the functionality of all the site services relevant for CMS and the capability to sustain the various CMS computing workflows at the required scale.

4.2.1 Sites

Every day the following quantities are monitored: the CMS Service Availability Monitoring (SAM) tests to check sites basic functionality and local CMS software and configuration; the success rate of the Job Robot, a load generator simulating user data analysis; the number and the quality of the data transfer links used in production by the site; the downtimes scheduled by the site. If any of the metrics based on the above information is not satisfied the site is declared in error state. A site is allowed to be in error state not more than two days over the last week, then it is declared "not ready". To recover from a "not ready state" the site needs to be ok for at least two consecutive days. In this way temporary glitches are allowed if recovered promptly.

4.2.2 Data Transfer Links

A site needs to have sufficient data transfer connections to other sites in order to perform CMS workflows. An ad-hoc task force (Debugging Data Transfers, DDT) [24] was created in 2007 to coordinate the debugging of data transfer links, in order to commission most crucial transfer routes among CMS Tiers by designing and enforcing a clear procedure to debug problematic links.

A LoadTest infrastructure was set up in a separate debug environment of PhEDEx to handle DDT test traffic. The task force was charged with scheduling LoadTest transfers, assisting site administrators with the configuration of data transfer middleware, troubleshooting transfer issues, and documenting common problems and solutions. The DDT procedures are now regularly used to certify links quality.

In order to pass commissioning criteria, a data link from Tier-0 or Tier-1s must demonstrate a rate of at least 20 MB/s over 24 h. Recognizing that uplinks from Tier-2 to Tier-1 sites have a lower requirement in the computing model, they are only required to transfer 5 MB/s. Links were routinely exercised and were only decommissioned if they failed to meet these requirements for three days in a row. Transfer quality on commissioned links is continuously monitored with low rate transfers.

All Tier-0/1 to Tier-1 links are currently commissioned. 37 Tier-2 sites have all of their downlinks from Tier-1 sites commissioned, and 2 more have seven out of eight links commissioned. 47 Tier-2 sites have at least two commissioned uplinks to Tier-1 sites.

4.3 Analysis Organization at CMS Sites

4.3.1 Organized Analysis Activities

In order to optimize the processing chain, CMS performs as many processing steps as possible in an organized way. Besides the re-processing on the raw data when improved reconstruction algorithms are available, a standard set of analysis passes is agreed with the physics groups and is performed promptly at the Tier-1 sites as the data arrive from the Tier-0 (or Tier-2 in case of simulated data). This step, known as skimming, performs data reduction both in terms of event size and number of events. The samples produced in this way are made available to the CMS physicists at the Tier-2s.

Normally only the team dealing with data processing operations (Data Operation) has access to the Tier-1 resources but in some special cases a physicist may need access to large samples of raw data, which can not be hosted at Tier-2's. Examples of this kind of activity are the detector

calibration or the Data Quality Monitor validation. A special VOMS role has been created to grant access to the Tier-1s to those physicists.

4.3.2 Physics Groups and Users Activities

Physicists are organized in Physics groups, each with its own internal organization. Each Physics group is associated with a few Tier-2 sites that support it by providing storage space and processing power. Each Tier-2 supports one or more Physics groups depending on its size. Currently, the association of Physics groups to sites is only reflected in policies for data location, but it is also foreseen to exploit VOMS groups to prioritize CPU usage at sites.

A nominal Tier-2 in 2008 had 200TB of disk-based storage. Making efficient use of the overall space is a challenging data management exercise. The current approach is decentralized administration with central oversight. The knowledge of the needs is aggregated in the group leaders who will decide which data should be hosted at Tier-2 for their community. To help them in the process, each Physics group is allocated a limited number of "quanta", each being currently 30 TB of disk and enough CPU to process it, hosted at few Tier-2s.

Each Tier-2 site also supports a local community of users, providing them with a common space for data of local interest and a Grid-enabled storage for each user where to e.g. receive output from CRAB jobs running at other sites. Each CMS user can access any data hosted at any Tier-2, but can only write at the Tier-2 that supports him as a local user.

The user-produced data stored at a Tier-2 can also be published into a local DBS in order to allow access to other colleagues. The data meant to be accessed by many users or transferred among sites, for instance the Physics Group specific event filtering and selections, could be exposed to the entire collaboration by registering it in Global DBS and performing their transfer with PhEDEx. User data are typically not produced with files large enough for wide area transfers or suitable for storage systems. Therefore a migration process involving the merge of the files and the validation of the data is foreseen before the registration in Global DBS and PhEDEx occurs.

4.3.3 Storage Hierarchy at Tier-2

In order to support all the functionalities required by the CMS model, the Tier-2 storage is divided into four logical parts:

- Local Group and User Space: roughly 30 TB for the local community and additional 1 TB per user.
- Physics Group Space: 60–90 TB of space is allocated to serve 2–3 Physics Groups. Representatives from the groups serve as data managers for the space and make subscription and deletion requests. The space for each group will increase with time as datasets grow.
- Centrally Controlled Space: 30 TB of space is identified at each Tier-2 under the central control of CMS. This is used to ensure that complete copies of the reconstruction datasets are available across the Tier-2s. This space can be further used as a flexible buffer to deal with operational issues and difficulties in the short term.
- Simulation Staging and Temporary Space: 20 TB is identified to stage simulation produced at the Tier-2s and other temporary files before they can be transferred to the permanent home.

4.4 User Support Model

User documentation is provided by a set of twiki pages composing the so called CMS Workbook. Documentation about the distributed environment and CRAB usage are available, as well as a troubleshooting guide. Tutorials including hands-on session are periodically organized. The day by day support is currently performed mainly via a mailing list (HyperNews) where users reports the problems they are facing. The reported problems range from problems with the infrastructure, site related issues to user's mistakes in tools

configuration or real bug report which are fed back to the developers. The CMS Savannah Computing Infrastructure portal is used to report problems across the distributed infrastructure such as data access problems, problems on specific CMS software versions at sites, etc.

5 Experience with CMS Distributed Analysis

Distributed Analysis has been ongoing for several years, passing through sequentially planned steps of increasing complexity, the so called data and physics challenges [10–12]. It has been used extensively during studies to prepare the CMS Physics Technical Design Report [25] and various detector commissioning activities. Last year's experience both in terms of dedicated commissioning tests and real users analysis is reported in the following sections.

5.1 Analysis Exercises Within Common Computing Readiness Challenge

During the Common Computing Readiness Challenge (CCRC08) in May 2008 various analysis exercises were performed to gain an overall understanding of the performance and readiness of the Tier-2 sites for CMS data analysis. Centrally organized job submissions were carried out both to understand the site performance characteristics and to exercise closely the kind of workflows expected by the physics groups.

Site performance measurement Different types of jobs, with increasing complexity, were used:

- long-running CPU intensive jobs with moderate I/O. This tests the basic submission and execution of an analysis job with no strenuous requirements on either submit rate, I/O, or stageout. The goal here was to fill all batch slots available for the analysis at a given site without stressing the site.
- long-running I/O intensive jobs provided some non negligible stress on the storage infrastructure at the sites.
- short-running jobs O(10 min) with local stage out of O(10 MB) file as output. These jobs run for a short time, with many jobs finishing per hour, thus leading to a significant write request rate at large sites.

Up to 40 sites were involved across EGEE, OSG, and NorduGrid. More than 100,000 jobs succeeded throughout the challenge. The error rates were very mixed, ranging from less than 1% at many sites to up to 50% at a few The failures were predominantly due to storage problems at the sites. In most but not all the cases, those problems were detected and fixed by the site administrators within 24 h. Ignoring the storage failures, the success rate in this exercise was found to be better than 99%. Overall success rate including the storage issues, ranged between 92–99% for these exercises.

Simulation of physics group workflows An exercise to mimic realistic physics group activities running on a list of associated Tier-2s was conducted. The CRAB server was used to submit realistic physics group tasks: analysis-like jobs reading a dataset at all sites and running for about 4 h with remote stageout of a O(20 MB) output file to a subset of Tier-2 sites. This simulates the computing model where each user has space at a Tier-2 while the datasets are generally distributed throughout the world. More than 100,000 jobs on about 30 sites were submitted in two weeks and the CRAB Server provided the needed functionality for job submission and tracking. Most failures were problems accessing the input data, from 0.1–10% up to 50% for pathological cases, and remote stageout issues were due to old Grid clients affecting from 100% to few % per site. These stageout issues were promptly fixed by the site administrators. During the second week, the number of sites with efficiency above 90% significantly increased, as shown in Fig. 4.

Chaotic job submissions People from Tier-2 sites were encouraged to submit jobs to other Tier-2 sites, to mimic a chaotic job submission pattern. This activity was clearly visible in the CMS Dash-

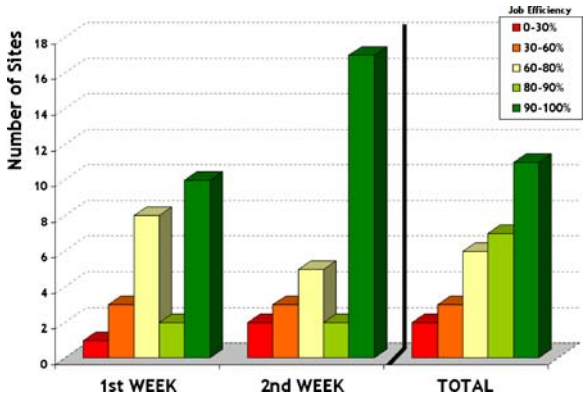

Fig. 4 Distribution of the job efficiency by site, when simulating physics groups workflows on CCRC08

board, showing lots of user activities at several sites. Figure 5 summarizes the number of active users per site, including Tier-1s, Tier-2s, Tier-3s and opportunistic usage. Around 60 sites participated in these tests.

The CCRC08 analysis challenge drove the level of activity and participation at Tier2s to an unprecedented scale in CMS.

5.2 CRAB Analysis Server Stress Test

In order to test the CRAB Analysis server scalability and reliability up to the expected CMS operational rates a dedicated test environment was set up in October 2008. The Storage Element, based on GridFTP, that CRAB server relies on was installed on a different machine with respect to the

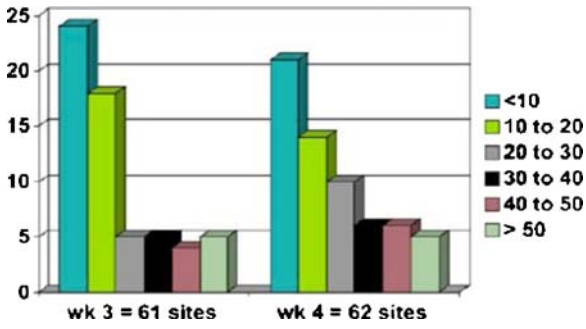

Fig. 5 Distribution of the number of users in a site, during the Chaotic phase in CCRC08

machine hosting the CRAB Server components. The aim was to decouple the load due the shipping of input/output sandboxes and the core workflow management. The machine hosting the Storage Element and the CRAB server were both 4 CPU 2,000 MHz Dual Core AMD Opterons with 4 GB RAM. The test was performed in the gLite context using two dedicated gLite-WMS. Monitoring information was collected from various sources like the CRAB Server database tracking the job flow and the CPU usage of its components, as well as from underlying services like MySQL and GridFTP server and gLite-WMS internal monitoring. The kind of jobs submitted were very short jobs running for less than a few minutes, not reading a dataset and without stage-out. This choice was made to provide a harsher environment for job handling due to higher rate of finished jobs and to limit the resource usage at sites.

Controlled submissions from different user certificates, thus emulating the CRAB Server usage in a multi-user environment, were performed. Different submission patterns were adopted by the 12 users involved. For example a user submitting 100 jobs every 15 min, another 500 jobs every 20 min, another 2,000 jobs every 6 h etc. plus a couple of users submitting randomly at their will. No CRAB Server misbehaviour was identified due to the multi-user environment. Jobs were submitted to more than 50 sites with a rate above 50,000 jobs/day. This helped identify some limitations in the communication between the components responsible for job output retrieval and handling that caused a backlog of jobs without output retrieved. This effect is more evident for homogeneous very short jobs, such as those used in the test, which have a higher finishing rate than real user's jobs which tend to finish at different times. Nonetheless, the backlog was absorbed in a few hours. This issue was taken into account in the development cycle and the code was optimized. About 120,000 jobs were successfully handled in 48 h, as shown in Fig. 6. The CPU load due to MySQL proved to be stable regardless of the database size increase with more jobs in the system. Overall the breakdown of CPU load usage is 2 CPUs for MySQL, about 1.5 CPUs for GridFTP

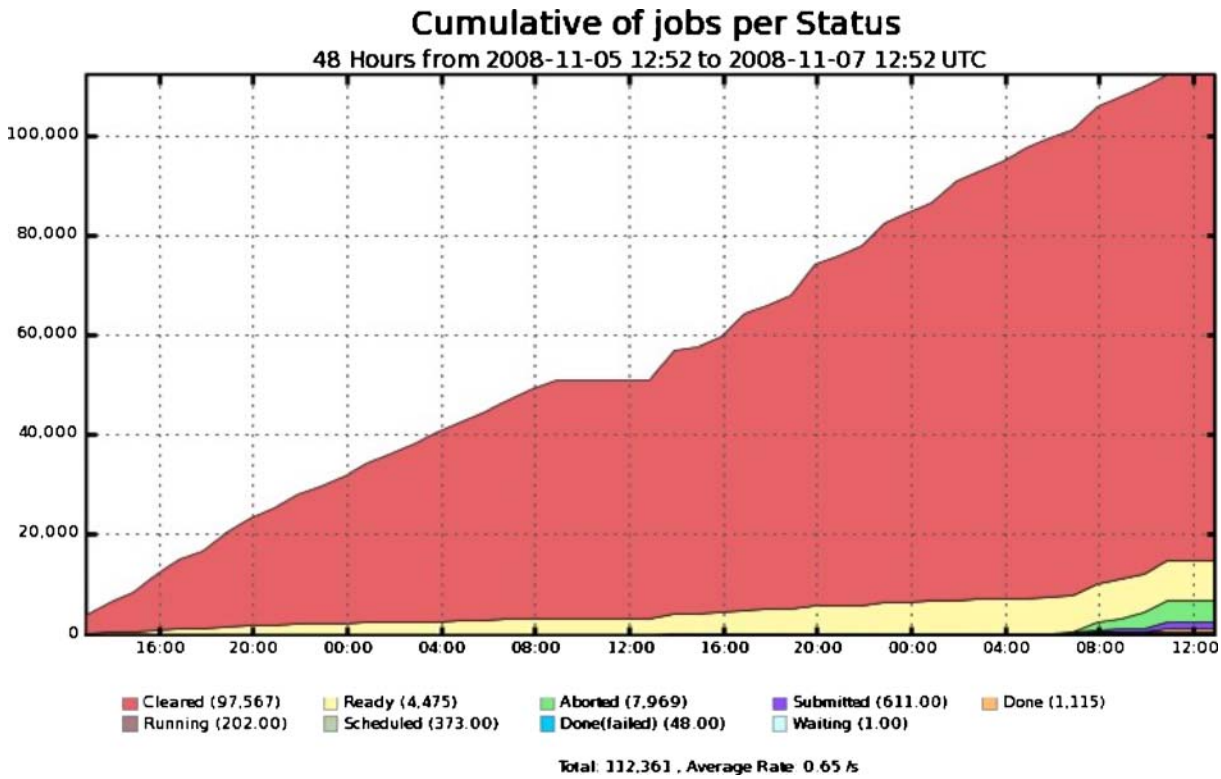

Fig. 6 Cumulative distribution of jobs submitted to CRAB Server during the multi-user test phase

and about 1 CPU for all the CRAB Server components, thus outlining the need of at least a 4 CPU machine. The load due to GridFTP is such that it's not compulsory to have the GridFTP server decoupled from the machine hosting the CRAB Server components.

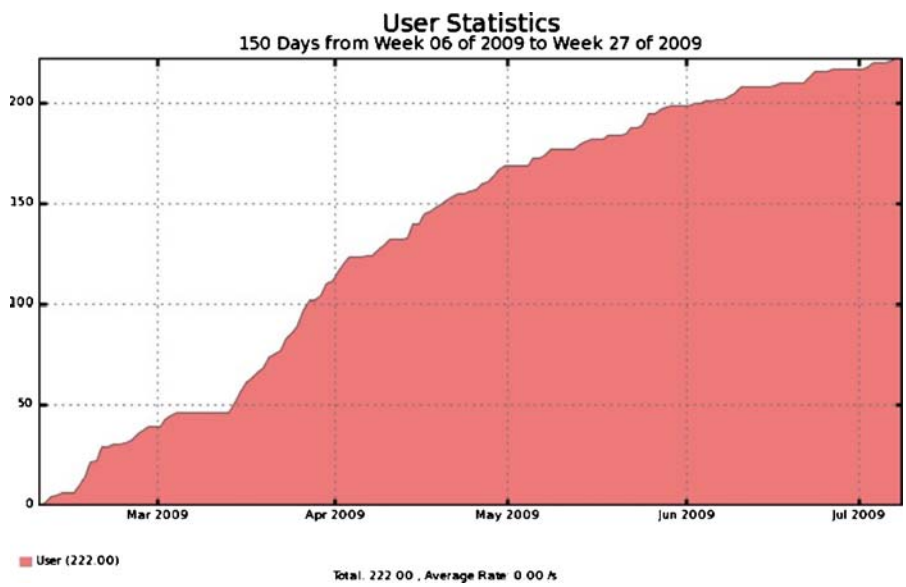

Fig. 7 Number of distinct users using a CRAB server instance during last 5 months

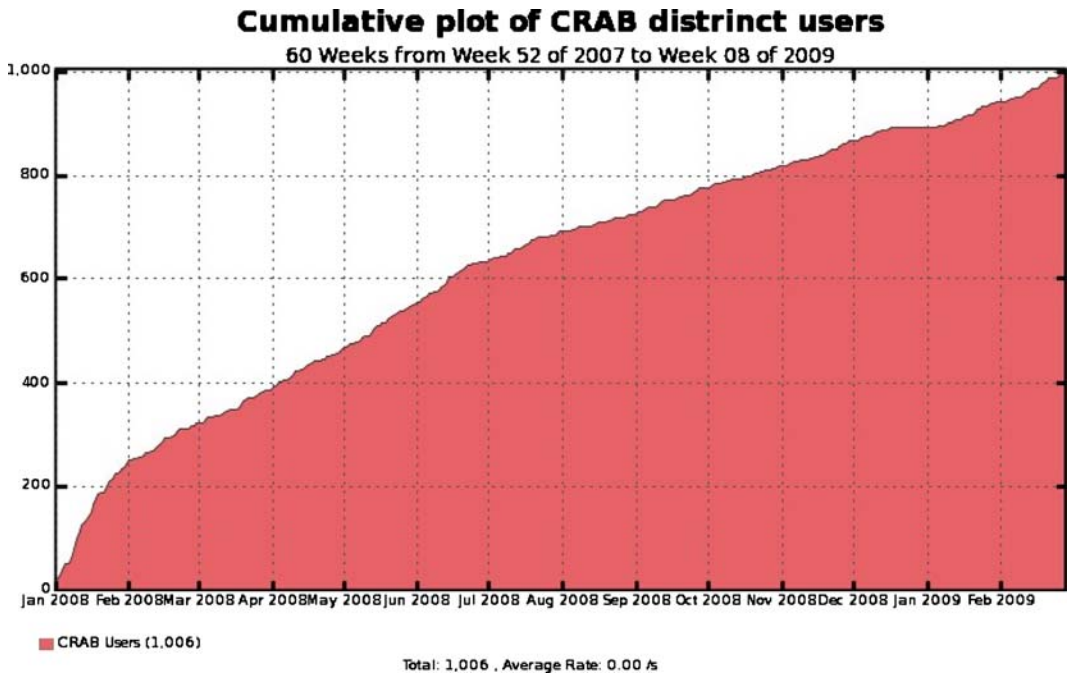

Fig. 8 Cumulative number of distinct CRAB users starting from 2008

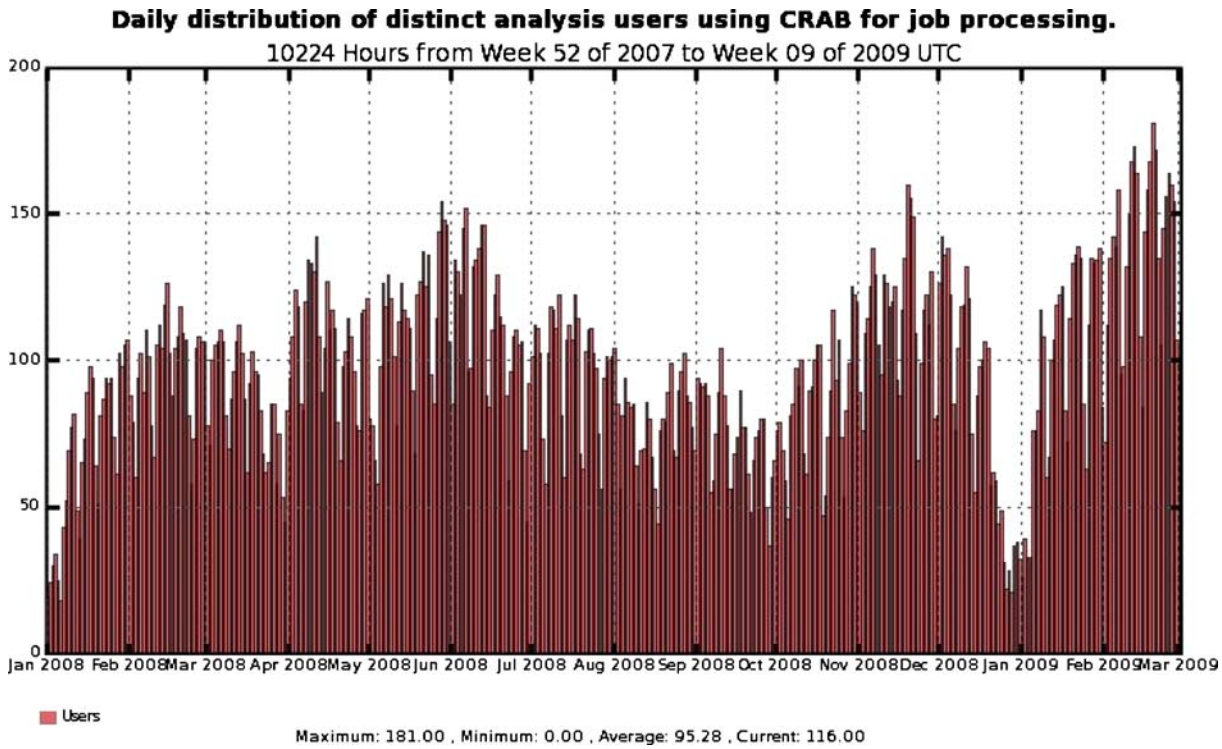

Fig. 9 Number of CRAB daily users in 2008

The number of users currently using a CRAB Analysis Server is significantly increased with respect to the stress test scale, e.g. the growing number of real users using a CRAB Analysis Server instance over the last 5 months is shown in Fig. 7. Currently the whole CMS analysis amounts to 30,000 jobs/day and the CMS Computing model expectation is to reach around 100,000–200,000 jobs/day. Some CRAB Server instances deployed at different sites to serve Physics group activities and a regional community, as foreseen, can cope with analysis needs.

5.3 Sustained Analysis

Distributed analysis is regularly performed by users for studies of the CMS physics discovery potential based on MC simulation and of the cosmic data collected in detector commissioning activities. The number of users is increasing over time. For instance since 2008 the number of distinct CRAB users has continuously grown to more than 1000, as shown in Fig. 8. This indicates a very broad usage of CRAB since it represents roughly 30% of the whole CMS community. The day by day distribution of CRAB users is shown in Fig. 9. An average of 95 different users per day use CRAB to submit their analysis jobs.

During the last year about 11 million analysis jobs were submitted. Peaks of more than 100,000 jobs per day have been reached, with an average of 30,000 jobs per day, as shown in Fig. 10. The distribution of analysis jobs at Tier-2s over the year is shown in Fig. 11. Current analysis activities has spontaneously surpassed, both in terms of number of jobs and number of sites, the scale reached in CCRC08 dedicated tests.

The distribution of analysis job efficiency over time is shown in Fig. 12. The average success rate is 61% with 4% of cancelled jobs, 10% of Grid failures and 25% application failures. Most

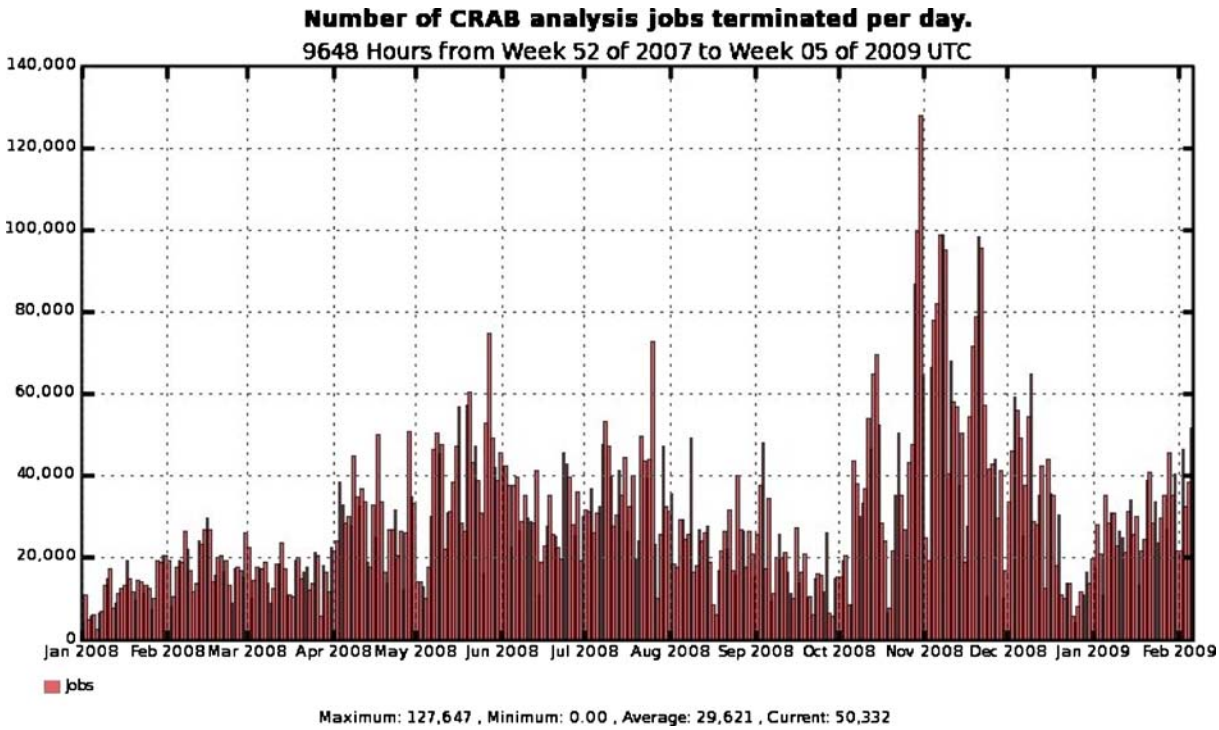

Fig. 10 Number of daily jobs terminated in 2008

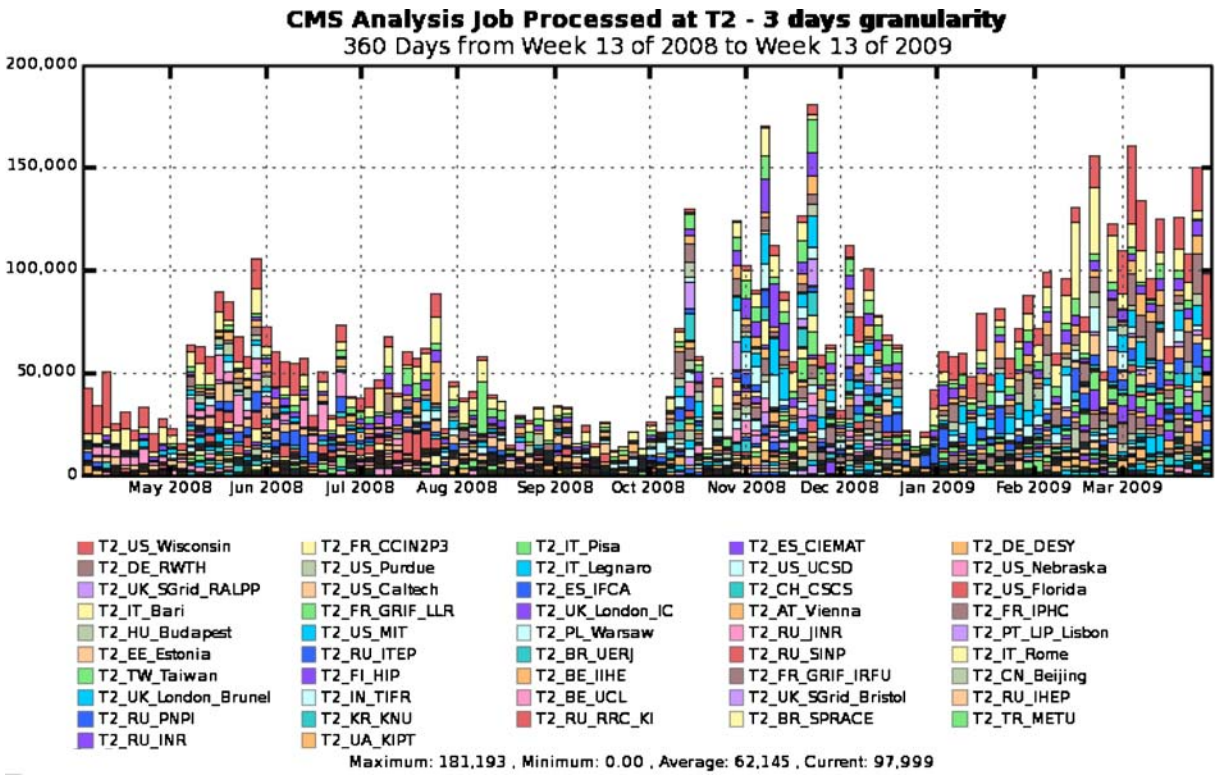

Fig. 11 Number of analysis jobs by Tier-2s during last year from CMS dashboard History view

of the failures faced by the users are due to remote stageout issues, user application errors and errors reading data at the site. Part of the failures are somehow expected, since analysis jobs run user code which may not have been thoroughly tested. For instance memory leaks or crashes in rare events might be hard to spot in the small scale test typically done by the users.

Failures in stage out of the data output files to remote storage can be due to users misconfiguring the remote destination or transfer problems. A hanging or failing stage out represents a waste of CPU since it occurs at the end of the job processing, so an approach to decouple the job processing and the remote stage out is under development. At job finishing the output will be stored locally at the site and the remote stage out will occur in an asynchronous step.

Failures accessing data at the site mainly expose problems with the storage at the site or inconsistencies between the data catalogue and what has been deleted at the site. Data consistency and integrity checks at all Tier-2 sites are performed periodically. These checks verify that the contents of the disks at the Tier-2 sites are consistent with the PhEDEx databases and DBS and reduce the rate of data access errors.

Grid failures are due to the underlying Grid system but also reflect site problems or jobs that spend too much time on the worker node and are killed by the local batch system, appearing as aborted by the Grid.

About 78% of the CMS analysis jobs were submitted using gLite-WMS. Since the gLite architecture is such that the system scales linearly with the number of gLite-WMSs used, analysis jobs are

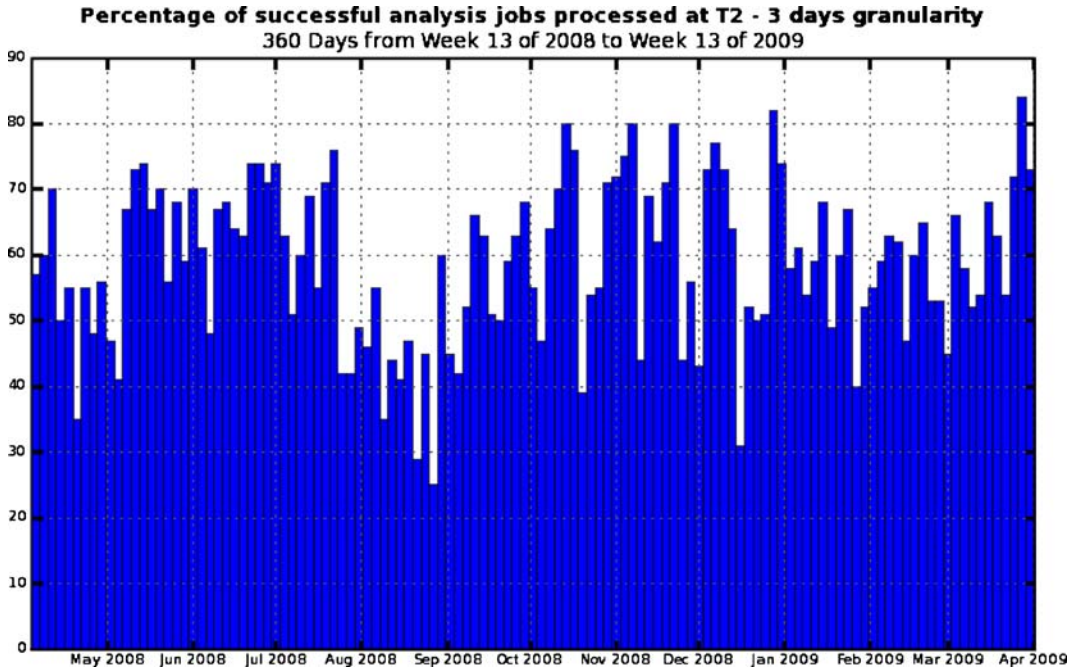

Fig. 12 Analysis job efficiency during last year

balanced currently over 7 WMS. The rest of the analysis jobs are submitted using Condor-G.

CRAB Analysis Server instances have been deployed in several countries (CERN, Italy, France). A couple of them are open to worldwide distributed CMS users. Other instances are being used by local communities or specific physics groups.

A complete example of analysis activity during the year has been the analysis of real cosmic data collected during a long period of data-taking, called CRAFT (Cosmics Run At Four Tesla) to commission the detector in preparation for LHC collisions. About 300 million events of cosmic muons were taken and their analysis is meant to

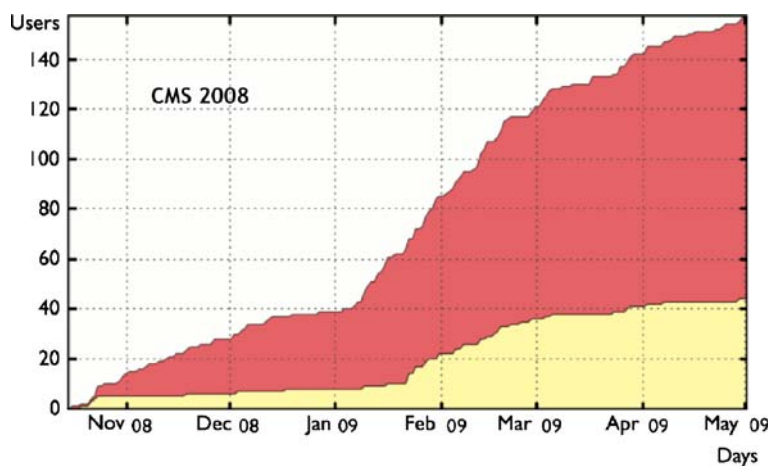

Fig. 13 Number of users analyzing CRAFT data using the CAF (*light color*) or the Tier-2s (*dark color*)

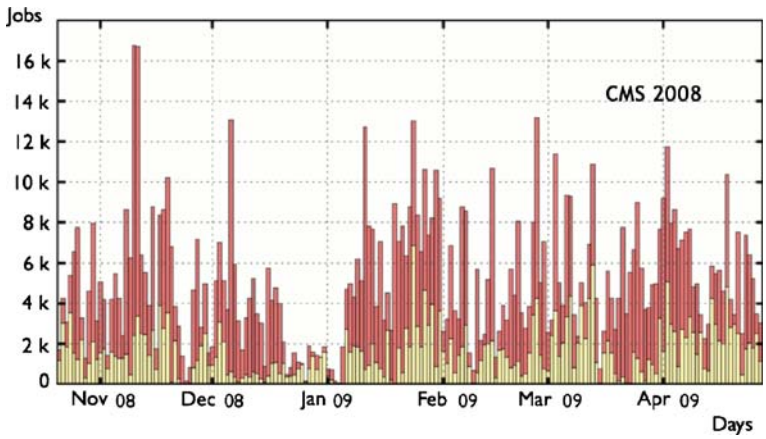

Fig. 14 Number of jobs analyzing CRAFT data using the CAF (*light color*) or the Tier-2s (*dark color*)

assess detector performance and perform physics studies. The raw data were transferred to Tier-1s where several re-processing and data skimming passes took place. The reconstructed data were all shipped to the CAF where calibration and alignment studies were performed in order to provide better conditions to be used in subsequent re-processing. Reconstructed data and their skims were also spread to Tier-2s, mainly to those associated with the Muon and Tracker groups with a total of about 20 sites hosting CRAFT data. The overall amount of CRAFT data transferred to Tier-2s was more than 300 TB. The number of users analysing CRAFT data during the year is shown in Fig. 13 where the breakdown of users using the CAF and those using the Tier-2s is also reported. The distribution of analysis jobs is shown in Fig. 14, roughly two thirds at Tier-2s and one third at the CAF. The job efficiency at Tier-2s is lower than that at CAF because, on top of the application errors, there are Grid failures and the stage out errors mentioned above.

Real user analysis jobs show worse job efficiency (around 60%) with respect to the efficiencies obtained during dedicated and controlled submissions such as computing challenge and Job Robot activities, described in Sections 5.1 and 4.2.1 respectively. Analysis specific operations are being defined to improve the success rate and user experience when utilizing the distributed computing environment and to ensure a functional system with a more efficient use of CMS resources on sustained user activities.

6 Related Work

The computing system presented in this paper is somehow similar to the systems developed by the other experiments at the LHC (ALICE [26], ATLAS [27], LHCb [28]), since they all share a similar environment and needs. They all rely on the Worldwide LHC Computing Grid (WLCG), a global collaboration linking Grid infrastructures and computer centres worldwide [22], which provides a globally distributed system for data storage and analysis.

The computing centres providing resources for WLCG are embedded in different operational Grid organizations across the world, in particular EGEE (Enabling Grids for E-SciencE, http://www.eu-egee.org/), OSG (Open Science Grid, [18]) and NDGF (Nordic Data Grid Facility, http://www.nordugrid.org/), but also several national and regional Grid structures.

EGEE, OSG and NDGF use their own Grid middleware distributions: gLite [29], VDT (http://vdt.cs.wisc.edu/) and ARC [20] respectively. Grid interoperation bridges the differences between the various Grids and enable the Grid Virtual Organizations for LHC experiments to access resources at the Institutions, independent of the Grid project affiliation.

Similarly to other LHC experiments, the CMS computing system implements a hierarchical ('tiered') infrastructure as first proposed by the MONARC project [30]. In CMS though, the

communication between Tier-2 and Tier-1 sites is not limited by region boundaries. The data distribution model in CMS is required to be under complete control of the operators and no automatic transfers triggered by middleware components are allowed. For this reason CMS developed the PhEDEx data distribution system that efficiently manages the operations among the almost complete mesh of links between sites and applies re-routing strategies for fault-tolerance and for optimization of the WAN traffic to by-pass temporary unavailabilities, make the optimal use of best performing links, etc. This adds value to the underlying layer based on FTS, which is used by all the experiments.

The Dataset Bookkeeping Service (DBS), based on a Oracle-RAC backend, has an implementation specific to CMS, as it happens for all other LHC experiments.

Since PhEDEx has the knowledge of the location of all CMS data, it also serves as a global data location catalogue. Using DBS and PhEDEx, a CMS user can find the list of logical file names of interest and the site where the files are hosted. The translation from logical file names to physical file names happens by means of a Trivial File Catalogue. This is different from what other experiments do, as they do such translation through a catalogue such as the LFC from the gLite middleware stack.

The CRAB system, the CMS job submission and control framework for analysis, is quite similar to other analysis frameworks used by LHC experiments such as GANGA [31], used by ATLAS and LHCb, and Alien [32], used by ALICE, but it is tailored to CMS needs. By means of specific plug-ins, CRAB can work in different environments, submitting jobs with workload management systems using "push-mode", such as the gLite-WMS [17] from the gLite stack, or using "pilot jobs", such as the glidein-WMS [19] based on Condor.

7 Conclusions

Commissioning a distributed analysis system of the scale required by CMS in terms of distribution and number of expected users is a unique challenge. In order to prepare for large scale physics analysis CMS has established a set of operations to extensively test all relevant aspects of the distributed infrastructure to support CMS workflows, such as performance and readiness of sites and transfer links. The Workload Management and Data Management components of the Computing Model are now well established and are constantly being exercised and improved through CMS-wide computing challenges, and first real cosmic data taking exercises, in preparation for the LHC collision data taking.

Acknowledgements We thank the technical and administrative staff of CERN and CMS Institutes, Tier-1 and Tier-2 centres and acknowledge their support. This work is co-funded by the European Commission through the EGEE-III project (www.eu-egee.org), contract number INFSO-RI-222667, and the results produced made use of the EGEE Grid infrastructure.

References

1. CMS Collaboration, Adolphi, R., et al.: The CMS experiment at the CERN LHC, JINST, 0803, S08004 (2008)
2. CMS Collaboration, CMS: The computing project. Technical design report, CERN-LHCC-2005-023, ISBN 92-9083-252-5 (2005)
3. Grandi, C., Stickland, D., Taylor, L., et al.: The CMS computing model, CERN-LHCC-2004-035/G-083 (2004)
4. Flix, J., Sciabà, A., et al.: The commissioning of CMS sites: improving the site reliability. In: Proceedings of 17th International Conference On Computing in High Energy Physics and Nuclear Physics. J. Phys.: Conf. Ser., in press (2009)
5. Flix, J., Sciabà, A., et al.: The commissioning of CMS computing centres in the worldwide LHC computing Grid. In: Conference Record N29-5 Session Grid Computing, Nuclear Science Symposium IEEE, Dresden (2008)
6. Afaq, A., et al.: The CMS dataset bookkeeping service. J. Phys. Conf. Ser. **119**, 072001 (2008)
7. Blumenfeld, B., Dykstra, D., Lueking, L., Wicklund, E.: CMS conditions data access using FroNTier. J. Phys. Conf. Ser. **119**, 072007 (2008)
8. Egeland, R., et al.: Data transfer infrastructure for CMS data taking. In: Proceedings of Science, PoS(ACAT08)033 (2008)
9. Tuura, L., et al.: Scaling CMS data transfer system for LHC start-up. J. Phys. Conf. Ser. **119**, 072030 (2008)
10. CMS collaboration: CMS computing, software and analysis challenge in 2006 (CSA06) Summary. CERN/LHCC 2007-010 (2007)

11. DeFilippis, N., et al.: The CMS analysis chain in a distributed environment. Nucl. Instrum. Methods **A559**, 38–42 (2006)
12. Fanfani, A., et al.: Distributed computing Grid experiences in CMS. IEEE Trans. Nucl. Sci. **52**, 884–890 (2005)
13. Bonacorsi, D., Bauerdick, L., on behalf of the CMS Collaboration: CMS results in the combined computing readiness challenge (CCRC08). Nucl. Phys., B Proc. Suppl. **197**, 99–108 (2009)
14. Evans, D., et al.: The CMS Monte Carlo production system: development and design. Nucl. Phys. Proc. Suppl. **177–178**, 285–286 (2008)
15. Codispoti, G., et al.: CRAB: a CMS application for distributed analysis. IEEE Trans. Nucl. Sci. **56**, 2850–2858 (2009)
16. Codispoti, G., et al.: Use of the gLite-WMS in CMS for production and analysis. In: Proceedings of 17th International Conference on Computing in High Energy Physics and Nuclear Physics. J. Phys. Conf. Ser., in press (2009)
17. Andreetto, P., et al.: The gLite workload management system. J. Phys. Conf. Ser. **119**, 062007 (2008)
18. Pordes, R., et al.: The open science Grid. J. Phys. Conf. Ser. **78**, 012057 (2007). http://www.opensciencegrid.org/
19. Sfiligoi, I., et al.: glideinWMS—a generic pilot-based workload management system. J. Phys. Conf. Ser. **119**, 062044 (2008)
20. Ellertet, M., et al.: Advanced resource connector middleware for lightweight computational Grids. Future Gener. Comput. Syst. **23**, 219–240 (2007). http://www.nordugrid.org/arc/
21. Andreeva, J., et al.: Dashboard for the LHC experiments. J. Phys. Conf. Ser. **119**, 062008 (2008)
22. LCG: LCG Computing Grid Technical Design Report, LCG-TDR-001 CERN/LHCC 2005-024. http://lcg.web.cern.ch/lcg/ (2005)
23. Bonacorsi, D., Egeland, R., Metson, S.: SiteDB: marshalling the people and resources available to CMS. In: Poster at the International Conference on Computing in High Energy and Nuclear Physics (CHEP 2009), Prague, 21–27 March 2009
24. Magini, N., et al.: The CMS data transfer test environment in preparation for LHC data taking. In: Conference Record N67-2 Session Applied Computing Techniques, Nuclear Science Symposium IEEE, Dresden (2008)
25. Bayatian, G.L., et al.: CMS technical design report volume II: physics performance. J. Phys., G Nucl. Part. Phys. **34**, 995–1579 (2007)
26. ALICE Collaboration: ALICE technical design report of the computing, CERN-LHCC-2005-018, ISBN 92-9083-247-9 (2005)
27. ATLAS Collaboration: ATLAS computing: technical design report, CERN-LHCC-2005-022, ISBN 92-9083-250-9 (2005)
28. LHCb Collaboration: LHCb TDR computing technical design report, CERN-LHCC-2005-019 (2005)
29. Laure, E., Fisher, S.M., Frohner, A., Grandi, C., Kunszt, P., et al.: Programming the Grid with gLite. Comput. Methods Sci. Technol. **12**(1), 33–45 (2006)
30. Aderholz, M., et al.: Models of networked analysis at regional centres for LHC experiments (MONARC). Phase 2 report, CERN/LCB 2000-001 (2000)
31. Moscicki, J.T., et al.: Ganga: a tool for computational-task management and easy access to Grid resources. Comput. Phys. Commun. **180**(11), 2303–2316 (2009)
32. Bagnasco, S., et al.: AliEn: ALICE environment on the Grid. Grid J. Phys.: Conf. Series **119**(6), 062012 (2008)