High availability through full redundancy of the CMS detector controls system

Gerry Bauer 6, Ulf Behrens 1, Matthew Bowen 2, James Branson 4, Sebastian Bukowiec 2, Sergio Cittolin 4, Jose Antonio Coarasa 2, Christian Del dicedque 2, Marc Dobson 2, Aymeric Dupont 2, Samim Erhan 3, Alexander Flossdorf 1, Dominique Gigi 2, Frank Glege 2, Robert Gomez-Reino 2, Christian Hartl 2, Jeroen Hegeman 2,7, Andre Holzner 4, Yi Ling Hwong 2, Lorenzo Masetti 2, Frans Meijers 2, Emilio Meschi 2, Remigius K. Mommsen 5, Vivian O’Dell 5, Luciano Orsini 2, Christoph Paus 6, Andrea Petrucci 2, Marco Pieri 4, Giovanni Polese 2, Attila Racz 2, Olivier Raginel 6, Hannes Sakulin 2, Matteo Sani 4, Christoph Schwick 2, Dennis Shpakov 5, Michal Simon 2, Andrei Cristian Spataru 2, Konstanty Sumorok 6

1 DESY, Hamburg, Germany
2 CERN, Geneva, Switzerland
3 University of California, Los Angeles, Los Angeles, California, USA
4 University of California, San Diego, San Diego, California, USA
5 FNAL, Chicago, Illinois, USA
6 Massachusetts Institute of Technology, Cambridge, Massachusetts, USA
E-mail: Giovanni.Polese@cern.ch

Abstract. The CMS detector control system (DCS) is responsible for controlling and monitoring the detector status and for the operation of all CMS sub detectors and infrastructure. This is required to ensure safe and efficient data taking so that high quality physics data can be recorded. The current system architecture is composed of more than 100 servers in order to provide the required processing resources. An optimization of the system software and hardware architecture is under development to ensure redundancy of all the controlled sub-systems and to reduce any downtime due to hardware or software failures. The new optimized structure is based mainly on powerful and highly reliable blade servers and makes use of a fully redundant approach, guaranteeing high availability and reliability. The analysis of the requirements, the challenges, the improvements and the optimized system architecture as well as its specific hardware and software solutions are presented.

1. Introduction
The Compact Muon Solenoid (CMS) [1] is a multi-purpose detector in operation at the Large Hadron Collider (LHC) at CERN, which is conceived to study a wide range of particles and physical phenomena produced in the high-energy collisions.

The CMS detector is composed of an inner silicon tracking system surrounded by a scintillating crystal electromagnetic calorimeter (ECAL), a brass/steel sampling hadron calorimeter, and

7 Now at Princeton University
a redundant muon system. The tracker and the majority of the calorimeters lie within the solenoidal magnet. The muon detectors and portions of HCAL lie outside in the iron yoke.

In order to control and monitor the detector status and the operation of all CMS subsystems, the CMS Detector Control System (DCS) [2] has been put in place. It is responsible for controlling and monitoring the global detector status and the operation of each detector involved in the data taking, and its infrastructure. This is required to ensure safe and efficient data taking so that high quality physics data can be recorded.

First, a short overview of the current architecture of the CMS DCS, its functionalities and performances during the first years of operation will be discussed. Then, the foreseen optimization of the software and hardware architecture will be presented, aimed to ensure redundancy in all the controlled sub-systems and to reduce any downtime due to hardware or software failures.

2. The CMS detector control system overview

The main aim of the CMS DCS is to ensure reliable operation and correct configuration of the detectors and equipment, providing control and monitoring over all sub-detectors, infrastructures and services involved in the experiment operation. In addition, it is in charge of the communication with the accelerator, with the Data Acquisition system (DAQ), and, during data taking, exchanges status and data with the Run Control (RCMS).

The DCS is responsible for the detector configuration during the start-up operation and for the data taking preparation of many DCS front-end components and their power supplies. This is required either to bring them into a running condition, according to the LHC states, or to simply define their running mode. Therefore, it must be able to respond to high level commands and translate them into the proper sequence of low level commands to the controlled hardware. It guarantees correct and effective operation, prevents human mistakes and increases the efficiency in executing routine actions.

Monitoring of the working conditions is another crucial task performed by the DCS. It provides both bookkeeping of detector and safety-related parameters, including alarm handling, and protecting critical components via a software access control. It provides the main interface for the experiment shifter to monitor and control the detector. This interface must be able to summarize effectively the status of the system, convey the information from millions of monitored parameters in an overall state, and be understandable by a non-expert operator. Moreover, selected data from DCS is exported to the CMS conditions database, which contains all the data describing the detector environment needed for the offline reconstruction, used for studying the detector response and for tuning its physical behaviour.

The DCS must be operational on a 24-hour basis during the entire year, and many of its features are needed at all times. To ensure this continuity, uninterruptible power supply (UPS) and redundant software and hardware systems are implemented in critical areas, however even non-critical nodes can be recovered in the order of minutes.

2.1. Architecture and Functionalities

The complexity and the criticality of the control tasks described impose several requirements on the control software. All of them are accomplished through a distributed Supervisory Control And Data Acquisition system (SCADA), running on PCs, in charge of acquiring the data from the front-end equipment and providing control functionalities, like handling commands, messages and alerts.

The instrumentation under the DCS responsibility consists of a wide variety of equipment, from simple front-end elements like sensors and actuators, up to complex computer systems that are connected to the SCADA stations through standard fieldbuses. For the DCS hardware readout, a set of industrial and custom drivers is used. OPC, standard solution for the industrial
power supplies, ELMBs, Siemens S7, SNMP, as well as other CERN custom drivers like DIP/DIM, are widely used across the system [2].

The supervisor functionalities provided by the DCS software are based on the commercial SCADA tool, PVSSII [3]. PVSS provides a distributed architecture where all the control processes, driver communications and monitoring tasks can be implemented in distributed processes, called managers, communicating over the network. Each control PC hosts a specific PVSS project that keeps a local runtime database holding the status of the monitored parameters. Several PVSS projects can be connected and exchange information, forming a distributed network of PVSS projects.

The supervisor architecture is developed in a tree-like Finite State Machine (FSM) node hierarchy. The top node, the Central DCS Supervisor, controls the single sub-detectors partition trees and interacts with the RCMS, as described in Fig. 1. The sub-detector controls, at the 2nd level of the hierarchy, represent the logical structure of the subdetectors, where commands flow down and states and alerts are propagated upwards. All the sub-detectors' control systems are integrated in a single control tree headed by the central DCS to ensure a homogeneous and coherent operation of the experiment. This allows the CMS DCS to be modular and partitionable in order to grant both independent control of individual sub-detectors or of a sub-detector partition, and the easy integration of new components.

In total the CMS DCS supervises \( \approx O(10^5) \) hardware channels, described by \( \approx O(10^6) \) parameters. The current system architecture is composed of about 100 servers, in order to provide the required processing resources. A rearrangement of the system software and hardware architecture is under development to ensure redundancy of all the controlled sub-systems and to reduce any downtime due to hardware or software failures. The new strategy, the motivations and improvements will be presented in the next sections.
3. Redundancy of the CMS Detector control system

In case of a failure of one of the system components, the current architecture of the CMS DCS ensures the possibility of re-establishing the normal operational condition in an acceptable amount of time. This is achieved thanks to the capacity in the CMS DCS of rebuilding every software component into a spare unit with automated procedures, requiring minimal expertise. Nevertheless, profiting from the enhancements in the computers technologies in the recent years and in the view of the improving system maintainability and reliability, a more redundant solution for the CMS DCS has been investigated. This approach helps decrease the possible downtime due to hardware failure, to minimize the maintenance of our components by benefiting from the latest hardware solutions available on the market.

The new optimized structure is mainly based on powerful and highly reliable blade servers and makes use of a fully redundant approach, guaranteeing high availability and reliability, through hardware and software components. The improvements in term of availability and reliability is accomplished by adding redundant components to the control system running the same software in parallel, in order to ensure correct operation of the protection and control system if one element fails. Redundant elements are therefore parts of the primary protection for a specific segment of the distributed system and are meant to improve system reliability.

3.1. Hardware redundancy

The entire CMS DCS server system has been rearranged in order to be fully redundant. The previous server architecture was based on a cluster of about 100 computers providing the required functionalities for the different subsystems, as described in tab. 1. These computers are located in the underground counting room and are based on rack-mounted computer servers. The high number of independent servers, each providing a different and eventually critical functionality, results in a system where a single failure can possibly lead to the impossibility of controlling one piece of hardware and induce a downtime in the operation.

The new configuration achieves high efficiency by adopting the blade system architecture for the entire DCS cluster. By adopting this technology with high-performance units, all the
Figure 3. Redundancy configuration of the CMS DCS computer architecture, connecting two different locations of the CMS experiment.

DCS functionality can be reproduced by reducing drastically the number of computing modules. However, this solution requires a partial reshuffle of the hardware and software components.

The adopted solution is based on Dell PowerEdgeM610 blade server, with dual 136 GB 15K RPM drives in RAID1, dual quad-core 2.33 GHz processors, 48 GB of RAM and four Gbit Ethernet ports. All the blade servers are located in a chassis, hosting up to 16 server blades. The redundancy is implemented at the level of single chassis components for most of the elements involved (i.e. cooling fans, power supplies, network switches, redundant memory with failover memory banks for memory mirroring, I/O modules) in order to minimize single points of failure. All components are shared through the midplane, which is passive to help ensure high reliability. The redundancy at the level of the power supply is accomplished in the chassis not only on the DC power unit (6 in a 3+3 configuration) but also at AC level. This is obtained connecting the chassis on different AC grids, in order to avoid the total loss of power in case one AC circuit or UPS fails. When a failover occurs, alert messages are sent to the chassis interface, as illustrated in fig. 2, and the resource rearrangement is completely handled by the chassis, without interrupting the service.

In addition, to ensure a higher level of redundancy, the new architecture foresees duplication of the DCS chassis in a different area of the experiment, in order to overcome a possible failure of the services or infrastructure at a single geographical area. In fact, the CMS computing infrastructure is distributed in two main areas: in the underground area, close to the experiment in a no-radiation hall, and in the surface building. In each location, two blade server chassis are installed and the two sites are connected with a redundant interconnection implemented with 4 routers [5], configured in failover mode, as shown in fig. 3. The two sets of chassis act as twin pairs, where for each blade installed underground an equivalent component, implementing the same functionalities, is installed on the surface and acting as an hot standby peer.

In order to work with the new architecture, the different front end connections have been changed to be network based avoiding any custom point-to-point connections. In fact, custom boards and commercial PCI cards, used in the previous configuration, have been replaced with
solutions based on Ethernet protocol.

A redundant configuration has been also applied to the storage elements needed for the DCS operation like the Network Attached Storage (NAS) and the CMS online database. The NAS is needed as a software repository and for configuration management, which are required by the DCS software for normal operation. In the same way, a specific configuration on the database side has been put in place using an Oracle RAC architecture [5] with 4 blades.

Table 1. The DCS size in term of monitored parameters. A comparison between the computer resources of the old and new configuration is also provided.

| System Name | Monitored Parameters | Controlled parameters | Number of PCs Old Configuration | Number of PCs New Configuration |
|-------------|----------------------|-----------------------|---------------------------------|---------------------------------|
| Tracker     | 350k                 | 20k                   | 14                              | 4                               |
| Calorimeter | 115k                 | 2k                    | 14                              | 4                               |
| Muon        | 435k                 | 30k                   | 30                              | 6                               |
| Trigger     | 1k                   | 0.5k                  | 2                               | 1                               |
| Alignment   | 3k                   | 0.5k                  | 3                               | 1                               |
| Services    | 20k                  | 1k                    | 35                              | 8                               |
| Total       | 934k                 | 34k                   | 98                              | 24                              |

3.2. Software Redundancy
The reorganization of all the software applications in a redundant way has represented an additional challenge. There are several advantages in adopting redundancy at the software level: the system can be kept alive and fully operational even in case of failures of part of the SCADA application on a single node, driver processes or operating system errors. Also, redundancy enables experts to perform operations, maintenance and upgrades on the single machine without interrupting the normal condition, just turning one node to passive.

The implementation of the software redundancy has been possible thanks to the built-in functionality provided by the SCADA application in use. CMS has been a leader among the LHC experiments in adopting this functionality for the control system.

PVSS allows users to duplicate a project on a secondary host. A redundant system consists of a pair of server nodes operating permanently with the same functional demands. One host acts as active peer and the other as hot standby. The secondary server synchronizes the data at runtime with the primary unit and is able to become active in case the primary fails, taking over the control. PVSS controls and ensures the automatic matching of the process image between the peers, the alarm data and data history as well as the system start-up condition. This is achieved through the two Redu Managers (Fig. 4), so both peers access the same system image (i.e., the values of all the variables describing the status of the controlled process). In this way, the redundant mechanism is mostly transparent for all the other managers, leaving to the active Event manager the handling of the access rights and priorities on the system image variable. In a redundant system each user interface is linked with both the active and passive system. Redundancy switching is thus smooth without impairing system operability.

The switch from one system to another is based on specific conditions continuously evaluated by PVSS and on error conditions with configurable weight and severity, e.g. manager status, TCP connections, drivers communication, hard disk capacity, available RAM memory, and operating system error messages (Fig. 5). In addition, custom conditions can be defined by the
user for determining the switch condition. Whenever a condition of failure or a change of status is detected, automatic switching is performed in the tenth-of-a-second range without data loss.

The reorganization of the computer resources have also imposed a general rearranging of the software resources to fully profit from the computational resources offered by the blade servers. The computer resources provided by a single blade unit are in fact about a factor 8 more powerful than the average computer resources previously used. Therefore, the number of server used has been reduced to one third, requiring a merging of several applications inside a single blade server. The merging of several components has been simplified by the mechanism for releasing and installing CMS DCS software. In fact, the CMS developers have been requested to provide their software application as autonomous plug-in components, totally independent from the environment where they are installed. These components are then distributed across the production SCADA projects according to a configuration database populated by control system administrators. This approach allows for versioning, load balancing and automatic recovery after PC failures, and allows reshuffling of the project functionalities without major redesign of individual components.

4. First operational experiences and perspectives

The hardware for the new configuration has been installed in its final location during the 2011 LHC winter shut-down. This project has been carried out since 2009 and, in the first phase, the functionalities provided by the CMS DCS as well as the possible hardware and software solutions to adopt (i.e. blades, hardware virtualization, operating system, and hardware control system elements) have been analysed, tested and validated.

The work has been accomplished adapting the different DCS functionalities to the new architecture without loosing backward compatibility. A commissioning and testing campaign has been performed on the new configuration to validate the pre-existing software in the redundant configuration, to analyse the possible limits and drawbacks and to optimize the performance of the merged configuration.

A special effort has been made in modifying the software components (i.e. framework [6], central and sub-detector applications) in order to make them able to work in a redundant environment. Moreover, all the external software and components not natively integrated in PVSS but part of the control system (i.e. FSM, access control, DIM, and DIP protocol), required specific changes and additional functionalities to operate in a redundant configuration.

Figure 4. Redundancy in PVSS. Commands and messages are handle by the Event and Redu manager, shared and distributed through the different managers.

Figure 5. Example of a typical redundant configuration in use on the production servers. Managers status, connection, error messages, and proprieties are presented.
Modifications were required not only to allow the coexistence at the same time of two running projects, but also to improve the switching timing from peers, which is strongly dependant on the particular components involved. Testing and validation has also been carried out on the DCS hardware readout in the redundant environment, to investigate the required functionalities and the impact of duplicating the systems involved in the hardware communication. A first redundant control project for specific DCS tasks was installed in production since 2010, partially using the configuration previously described (Fig. 5). The migration of the current system on blade servers started on January 2012 and, up to now, about 15% of the system has been migrated. It has been proven to work properly in the production system during this year. The migration will be completed during the 2013 LHC shut-down. Major challenges are expected to be the migration of the single sub-detectors' control system where performances and driver communication play a crucial role. The identification of the performance and the optimization of the configuration will be part of the upcoming studies, since the merged system will handle a number of monitoring elements three times bigger than the current configuration.

5. Summary
The new redundant architecture for the CMS DCS has been presented. A comparison with the previous architecture has been provided, underlying the advantages and the improvements gained in term of reliability and system dependency. The redundant hardware and software structures has been described, motivating the components selection and the performances obtained. Finally, the current status of the project, the problems faced in the first phases and the solution adopted to overcome them has been described, as well as the roadmap and challenges to deal with in the view of the full system migration in 2013.

References
[1] Adolphi R et al. (CMS Collaboration) The CMS experiment at the CERN LHC2008 JINST 3 S08004
[2] Gomez-Reino R et al. 2005 CMS DCS Design Concepts (Geneva, Switzerland: ICALEPCS)
[3] PVSSII, now available as SIMATIC WinCC Open Architecture, http://www.pvss.at/index.asp.
[4] B. Franek and C. Gaspar , “SMI++: Object oriented framework for designing and implementing distributed control systems”, presented at 2004 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS / MIC), Rome, Italy, 16–22 Oct. 2004.
[5] The CMS online cluster: IT for a large data acquisition and control cluster G Bauer et al 2010 J. Phys.: Conf. Ser. 219 022002
[6] M. Beharrell et al. , “Technology Integration in the LHC experiments Joint COntrols Project, CHEP01, Beijing, September 2001. http://cern.ch/itco/Projects-Services/JCOP/.