HANDLING LARGE DATA AMOUNTS IN ALICE DCS

Peter Chochula[(1)], André Augustinus[(1)], Vladimír Fekete[(2)], Lennart Jirdén[(1)], Svetozár Kapusta[(1), (2)], Peter Rosinský[(1), (2)], [(1)]CERN, Geneva, Switzerland,
[(2)]Comenius University Bratislava, Slovakia.

Abstract

The amount of control data to be handled by the ALICE experiment at CERN is by a magnitude larger than in previous-generation experiments. Some 18 detectors, 150 subsystems, and 100,000 control channels need to be configured, controlled, and archived in normal operation. During the configuration phase several Gigabytes of data are written to devices, and during stable operations some 1,000 values per second are written to archival. The peak load for the archival is estimated to 150,000 changes/s. Data is also continuously exchanged with several external systems, and the system should be able to operate unattended and fully independent from any external resources. Much care has been taken in the design to fulfill the requirements, and this paper will describe the solutions implemented. The data flow and the various components will be described as well as the data exchange mechanisms and the interfaces to the external systems. Some emphasis will also be given to data reduction and filtering mechanisms that have been implemented in order to keep the archive within maintainable margins.

INTRODUCTION

The primary task of the ALICE Detector Control System (DCS) [1] is to ensure a safe and correct operation of the ALICE experiment at CERN. It is in charge of configuration, control and monitoring of 18 ALICE sub-detectors, their sub-systems (Low Voltage, High Voltage, etc.) and interfaces with various services (such as cooling, safety, gas, etc.). The operation of the DCS is synchronized with the other ALICE online systems, namely the Data Acquisition System (DAQ), the Trigger (TRG) and the High Level Trigger (HLT) through a controls layer: the so-called Experiment Control System (ECS) as shown in Figure 1.

The core of the ALICE DCS is a commercial SCADA System PVSSII. This is built as a collection of autonomous software modules (managers) which communicate via TCP/IP. About 90 individual PVSSII systems consisting of 900 managers are running on 150 computers. Together they form one global distributed system. A number of tools developed within the Joint Controls Project (JCOP) and by the ALICE collaboration is provided to extend the PVSSII capabilities [2].

The data flow in the ALICE DCS covers two classes of data:

- The synchronization data that allows for coherent operation of all DCS sub-systems and assures coordination between online systems. It consists of commands and states transferred between different components of the control system.
- The controls data which includes all information needed to configure the detectors and the control system itself. It also carries all information read back from detectors.

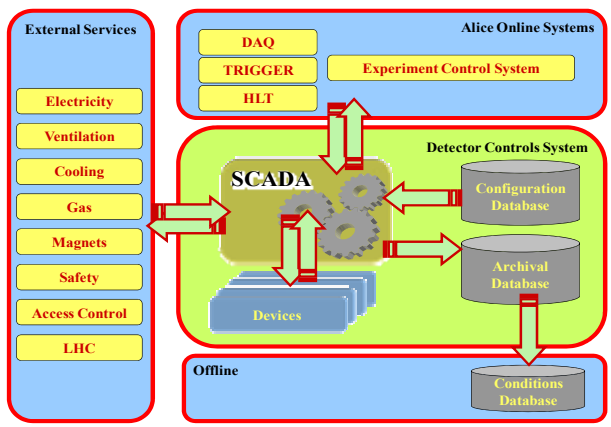

Figure 1: The ALICE DCS in context with the online systems.

THE SYNCHRONIZATION DATA FLOW

In order to allow for coherent and parallel operation of many devices with different operational requirements, the DCS is internally grouped in logical blocks. Each sub-detector is treated as an autonomous entity which reacts to commands and publishes states reflecting its own internal operation [3].

At the sub-detector level, the DCS is divided into sub-systems, representing control objects such as Low Voltage (LV), High Voltage (HV), Front-end and Readout Electronics (FERO), Gas, Cooling, etc.

Each subsystem consists of devices built from modules. The smallest controllable unit is a device channel, which can be a voltage channel, a temperature probe, a register of a front-end chip, etc. There are 150 subsystems in total which form the ALICE DCS.

The units are organized in a hierarchical way. Upper layers provide a simplified view of the state of underlying layers. At the top level, the global DCS object represents the whole control system. It interacts with ECS in order to synchronize with the other online systems.

The communication between units is handled by the Finite State Machine (FSM) mechanism, implemented in the SMI++ environment [4]. The behaviour of each unit is modeled as a finite state machine which reacts to commands sent by its parent and reports back its own internal status. Standard state diagrams are used across

the whole DCS and provide an abstraction level for controlled units. For example the same commands, such as OFF or ON, are recognized by all power supplies, even if these are produced by different manufacturers and have different control interfaces.

THE CONTROL DATA FLOW

The DCS control data consists of configuration data stored in the Configuration Database and monitoring data retrieved from the detectors and devices. Configuration data contains all information needed for the detector operation, such as settings for all controlled devices, operational limits, alert limits, archival settings, readout refresh rates, etc.

Once configured, all devices are controlled and monitored and the resulting data is stored in the DCS Archive. If any of the monitored parameters exceed a predefined range, the DCS can take automatic corrective actions adjusting the device settings or initiate a software interlock to protect the equipment.

Each archived value is tagged by a timestamp which indicates the time of the acquisition. If needed, this can be correlated with external events. For this reason the DCS also archives data arriving from external systems. All data stored in the archive is available for display and analysis using graphical User Interfaces (UI). A subset of this data is also transferred to the Offline Conditions Database at the end of each run for use in physics data analysis.

The HV and LV subsystems consist of 270 power supply crates with about 4000 controlled channels. For each channel 20-30 parameters are monitored and controlled. For example, a typical low voltage channel provides values of its voltage and current. In addition, the controls system must configure the desired set value, measurement refresh rate, alert limits and corresponding software actions, archival settings (such as smoothing parameters) etc.

Probably the most challenging part in terms of dataflow is the Front-End Readout Electronics (FERO) [5]. The DCS continuously monitors 30000 channels, in addition 70000 channels are accessible via DCS and can be read out on operator's requests (i.e. for debugging purposes). Many of these channels provide several parameters to DCS. In total about 160 MB of configuration data is loaded from the database into the FERO chips. In addition, up to 6GB of data created dynamically each run (such as pedestals) is loaded into the electronics via DCS.

The communication with the front-end chips is established either directly from the control computer farm using a dedicated bus, or over the network using 800 single-chip computers mounted directly on the front-end modules. For example, the TPC detector has 216 computers talking to 4356 front-end cards and 34848 chips. About 3.7 million registers and 558 000 pedestal memories (10 bit) need to be programmed by the DCS for this detector.

A large variety of access technologies is deployed to communicate with the devices. At the field layer the DCS communicates over several buses including JTAG, VME, CANbus, Profibus, RS-232, Ethernet, EasyNet, etc. Deployment of hardware abstraction layers significantly reduces the complexity of communication mechanisms between the SCADA system and devices. Commercial devices, such as power supplies, are interfaced to PVSSII via the OPC mechanism which is an industrial standard. Another layer, the FED server, hides the complexity of the various front-end architectures used in ALICE. The FED API provides an uniform method for accessing the hardware and is used to access all architectures controlled by DCS.

A number of additional devices such as temperature probes, pressure and flow sensors, NMR probes, etc. are needed to operate ALICE. Data from these devices together with data flowing from external services is processed by DCS. In total about one million parameters are needed to control the ALICE Experiment.

THE DCS DATABASE SERVICES

All PVSSII systems use a common database for archival. This is implemented as an ORACLE Real Application Cluster (RAC) consisting of 6 database server nodes and 3 redundant SAN disk arrays providing total storage capacity of 24 TB. The same RAC is used to store configuration data of the Front-End Electronics and the devices. Most of the DCS channels are monitored at around 1Hz refresh rate and are archived. To keep the database size within reasonable limits, data compression is applied at several stages of the data acquisition and processing.

The device channels are typically polled at frequencies of 1-2Hz. A deadband mechanism applied within the drivers. Each readout value is compared with the previous measurement and is injected into PVSSII only if the difference exceeds the predefined threshold as shown in Figure 2. This mechanism reduces the traffic between hardware and PVSSII to 0.1Hz. Additional smoothing based on the same principle is applied at the level of the PVSSII archive managers.

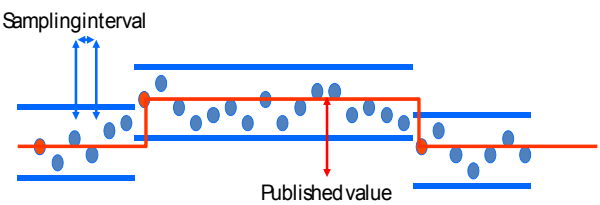

Figure 2: Principle of the data reduction mechanism implemented in FED servers and archive managers.

The steady archival rate for ALICE is estimated to 1000 inserts/s throughout the year. The database service is designed to cope with a steady rate of 150000 inserts/s,

which corresponds to a peak load during the ramp-up periods, when most of the monitored channels change.

Roughly 250 MB of configuration data is loaded from the database and written into the ALICE devices. A versioning system assures that there is a configuration version available for each ALICE running mode (cosmics, calibration, physics with protons or ions…) and limits the data duplication.

The expected total size of configuration and archival data needed for one year of running is ~20TB, this includes also online backups for fast recovery.

THE CONTROL DATA EXCHANGE WITH SYSTEMS EXTERNAL TO THE DCS

The DCS is designed for 24/7 operation and is able to work in stand-alone mode. However, during normal experiment operation a significant amount of controls data needs to be exchanged with the offline, HLT, DAQ and LHC Machine. There are four mechanisms implemented to achieve these tasks:

- The data required for the offline reconstruction is stored in the offline conditions database. A mechanism based on an AMANDA server has been developed. The AMANDA server receives requests from an offline client and retrieves the data from the archive. The results are formatted into blocks and sent to the client to be stored in the conditions database. It is expected that the AMANDA server will transmit 60MB of data at the end of each run to the offline. However, this requires fine-tuning of the archival and data smoothing. In the beginning of the ALICE operation these numbers can be significantly higher.
- In addition to offline, the same AMANDA server-client mechanism is also used to transmit data to the HLT.
- File Exchange Servers (FXS) are used to transfer huge amounts of data. This includes for example pedestals computed by DAQ or configuration parameters prepared on the HLT farm. Some parameters, like images produced by alignment systems, are processed by DCS and sent to offline through a dedicated FXS.
- A small amount of data is exchanged directly via a mechanism based on the DIM protocol [6]. The DCS publishes some parameters via DIM servers. This is used for example to transmit slowly changing detector parameters to HLT or to read back the HLT farm status and to communicate with services and the LHC accelerator.

CONCLUSIONS

The dataflow in the ALICE Control System has been carefully studied and tested. The scale of the system and the level of distribution have been adjusted according to the obtained results.

The database service is a vital component of the dataflow and much emphasis was put into its definition. The studies have led to the selection of technology based on ORACLE RAC which allows for efficient load balancing and provides the necessary level of scalability for future upgrades.

REFERENCES

[1] ALICE TDR of the Trigger, Data Acquisition, High-Level Trigger and Control System, CERN-LHCC-2003-062.

[2] A. Augustinus et al., The ALICE Controls System : a Technical and Managerial Challenge, 9th International Conference on Accelerator and Large Experimental Physics Control Systems ICALEPCS 2003 , Gyeongju, Korea , 13 - 17 Oct 2003.

[3] G. de Cataldo et al., Finite state machines for integration and control in ALICE, ICALEPCS 2007

[4] C. Gaspar et. al., SMI++ Object Oriented framework for designing Distributed Control Systems (ps) Presented at: Xth IEEE Real Time Conference 97 (Beaune, France, Sep 22-26 1997.

[5] P.Chochula et al.,Control and Monitoring of Front-End Electronics in ALICE, 9th Workshop on Electronics for LHC Experiments LECC 2003 , Amsterdam, The Netherlands.

[6] C. Gaspar et al., DIM, a Portable, Light Weight Package for Information Publishing, Data Transfer and Inter-process Communication (pdf) Presented at: International Conference on Computing in High Energy and Nuclear Physics (Padova, Italy, 1-11 February 2000).