COMPUTING ARCHITECTURE OF THE ALICE DETECTOR CONTROL SYSTEM

A. Augustinus, P. Chochula, L. S. Jirdén, M. Lechman, P. Rosinský,
CERN, Geneva, Switzerland
O. Pinazza, INFN Sezione di Bologna, Bologna, Italy and CERN
G. De Cataldo, INFN Sezione di Bari, Bari, Italy and CERN
A. N. Kurepin, Institute for Nuclear Research of the Russian Academy of Sciences, Moscow, Russia
A. Moreno, Universidad Politécnica de Madrid, ETSI Industriales, Madrid, Spain

Abstract

The ALICE Detector Control System (DCS) is based on a commercial SCADA product, running on a large Windows computer cluster. It communicates with about 1200 network attached devices to assure safe and stable operation of the experiment. In the presentation we focus on the design of the ALICE DCS computer systems. We describe the management of data flow, mechanisms for handling the large data amounts and information exchange with external systems. One of the key operational requirements is an intuitive, error proof and robust user interface allowing for simple operation of the experiment. At the same time the typical operator task, like trending or routine checks of the devices, must be decoupled from the automated operation in order to prevent overload of critical parts of the system. All these requirements must be implemented in an environment with strict security requirements. In the presentation we explain how these demands affected the architecture of the ALICE DCS.

ALICE DCS OVERVIEW

The mission of the Detector Control System (DCS) [1] of ALICE experiment is to provide an overall supervision of the experimental apparatus, ensuring correct and safe operation during the physics data taking and also during the standby periods.

Operating in a continuous 24/7 mode most of the year, the DCS performs hierarchical control of the 20 subdectors of ALICE, supervises common systems and infastructure services and communicates with external systems. Full remote control and monitoring is required for the devices located in the underground areas inaccessible during the beam time.

The ALICE DCS uses a variety of devices supervised by the SCADA software. The devices communicate with the supervisory layer using either ethernet networks, or industrial fieldbuses. Distributed and hierarchically designed supervisory system consists of control applications developed using a commercial system (PVSS) and CERN-developed tools.

User interfaces (UI) at different levels provide experts and operators with convenient control panels, allowing ALICE to be routinely operated by one shifter.

OVERALL DCS DESIGN

The DCS partitioning and control hierarchy follows the logical structure of the experiment. There are 20 subdetectors of different complexities and sizes, ranging from TPC and TRD (10 supervisory control nodes) down to ACO or ZDC with just one control computer.

There are several non-detector projects that provide centralized services (rack control, spaceframe monitoring, access control, global variables, DIM server) or communicate with the external systems (electricity, cooling, ventilation, magnet control, environment monitoring, radiation monitoring detector safety system, LHC services, gas control).

The field layer of the DCS consists of many different types of devices - power supplies (HV and LV supplies of several manufacturers), VME processors, custom made front-end DCS boards, TELL boards used in LHC-related projects, ELMB boards used mainly for the monitoring, PLC controllers and other commercially available or custom made equipment. While a large majority of the devices communicate via Ethernet (processors, boards featuring embedded Linux, but also a majority of the power supplies), there are also several industrial buses in use (CANbus, Profibus, Modbus, RS232, VME/VXI, JTAG).

Most widely adopted middleware communication protocols (running on top of TCP/IP) in ALICE DCS are OPC (industry standard, provides useful smoothing and data reduction) and DIM [2] (CERN-developed protocol - used in the front-end communication, state machine message passing, as well as for the communication with external systems).

The supervisory layer is based on commercial SCADA software (PVSS-II from ETM [3]) and JCOP framework [4] running under Windows (XP and Server 2003). Detector DCS experts developed the control applications for the particular subdector with the support of a small central DCS team that is also responsible for all the central DCS systems and infrastructure.

The front-end DCS boards are controlled by Linux-based applications based on a custom framework developed in ALICE. They communicate with their corresponding PVSS supervisory project by DIM.

The control hierarchy is built by the detector and subsystem developers using the finite state machine mechanism of the FSM package [5] developed at CERN. It allows to create a multilayer hierarchical tree structure distributed over several PVSS projects and control hosts. The detector top nodes are integrated into the overall ALICE control flow supervised by the central ALICE DCS node, which allows to bring the whole detector to the desired state (e.g. READY, STANDBY, OFF) by a single FSM command given at the top. At the various levels, the subsystem's FSM logic agent determines which actions should be executed for a given command and distributes the appropriate commands to the child nodes. It also calculates it's own state from the states of its sub-nodes and reports it to the parent node.

The user interfaces built on top of the control applications (based on PVSS and JCOP framework) allow experts and detector operators to bring the whole detector or its part to a desired state by one mouseclick. One top-level command creates an avalanche of commands propagated down the tree and the flow of corresponding state changes backwards. FSM inclusion/exclusion mechanism combined with the PVSS access control capabilities ensure proper ownership of a given sub-tree (either in the central tree or excluded for the experts).

The alert system based on PVSS datapoint properties and JCOP framework tools provides effective means to detect anomalous conditions so that the operators and experts can efficiently react and bring the detector or a subsystem back to the full health.

DCS COMPUTING

The design of the DCS computing cluster follows the overall DCS partitioning (detectors, central services, external interfaces).

Each detector has at least one PVSS worker node, one operator node running the user interface and most of the detectors have also a dedicated Linux front-end node. Larger detectors distribute the control tasks over several PVSS hosts according to the functionality and/or device types (HV, LV, gas, cooling, front-end) that can be further subdivided into the groups according to the detector layout (side, sector).

The detector top node runs the upper FSM control layer(s), distributes commands, collects states of the various subsystems and communicates with the top ALICE DCS node.

The operator nodes are dedicated for the experts to run their (often CPU and memory hungry) user interfaces independently so that they do not affect the main control applications running on the worker nodes.

Several common projects are running on the dedicated central PVSS nodes, such as rack control, gas control, spaceframe monitoring, global variables, alerts or PVSS access control.

The LHC interface subsystem consists of PVSS hosts supervising tasks like beam conditions and luminosity monitoring or beam injection handshake procedure with the central LHC control centre.

Apart from the described FSM control flow (commands and states) there's much larger flow of the configuration and monitoring data in the ALICE DCS hierarchy (peak rate of 150 000 changes/sec).

The configuration data (originating from the database and custom-formatted files) are sent at the various stages of the setup phase from the control layer to the devices (most of the load goes to the front-end). The monitoring data coming from the devices (via OPC, drivers and PVSS managers) are processed at several levels of the DCS hierarchy using PVSS datapoint objects and can be archived to the database by a specialized PVSS DB manager (about 1000 changes/sec archived). The total size of the data stored in the database is at the level of 20 TB/year (configuration plus archive).

The Linux-based ORACLE database system (housing the configuration database and the archive) is also part of the on-site DCS cluster, as well as several fileservers, bootservers, engineering nodes and other support systems.

Several Windows-based fileservers are used to store and backup the DCS projects, software repository and various tools. Central runtime fileserver hosts certain part of all PVSS projects (such as panels and scripts). Several Linux-based bootservers and fileservers support diskless systems (VME processors, boards based on embedded Linux).

The monitoring of the DCS hierarchy is performed at the level of PVSS (JCOP framework tools), the cluster monitoring is mostly based on Microsoft system monitoring package (SCOM) and Intel toolkit (ISM). In total we have over 200 control system computers, about 100 of them running PVSS applications, 70 serving detector front-ends and non-detector services, the rest belongs to the central services (database, fileservers, gateways etc.).

ALICE DCS NETWORK

Due to the nature of the experimental environment we decided to run the DCS on a private network decoupled from the CERN General Purpose Network (GPN).

Many networked devices used in the DCS lack important security features, others are difficult to update/upgrade. Even if the patches/updates exist they have to be first properly tested, that cannot be fully achieved in the limited lab test setups. The requirement of a continuous and stable operation also disfavors frequent changes at any level.

A CERN-wide policy for the computing and network infrastructure for controls (CNIC) [6] was proposed and implemented that allows to restrict the traffic between the security domains (ALICE, GPN, TN). Only certain vital services running in the CERN computing centre are visible from the ALICE network, as well as those provided by the LHC groups on the Technical network (TN) which is isolated from the GPN as well. The networking infrastructure (routers, switches) and tools (to

create and apply the rules) are provided by CERN IT department. It is also possible to further protect certain sensitive devices (e.g. PLCs) within a security domain, isolating them from all the DCS networked nodes but the few explicitly permitted. The data transfer to and from the DCS network is performed via a dedicated gateway, using the fileservers running inside the two security domains.

Only the central and detector DCS experts and developers are authorized to access the DCS network via an application gateway.

The network is physically divided into several starpoints provided by CERN IT that cover all experimental areas including the counting rooms and experimental cavern. In total we have over 1200 networked devices (600 DCS boards, 250 computers, 200 power supplies).

REFERENCES

[1] P.Chochula, A.Augustinus, L.Jirden, S.Kapusta, P.Rosinsky, "Handling large amounts of data in ALICE", ICALEPCS07, Knoxville, USA 2007.

[2] C. Gaspar,M. Donszelmann, "DIM - A Distributed Information Management System for the DELPHI Experiment at CERN", Proceedings of the 8th Conference on Real-Time Computer applications in Nuclear, Particle and Plasma Physics, Vancouver, Canada, June 1993.

[3] ETM website http://www.etm.at (the product was recently rebranded as "Simatic WinCC Open Architecture")

[4] G D.R.Meyers, "The LHC experiments Joint Control Project, JCOP", ICALEPCS99, Trieste 1999

[5] B.Franek, C.Gaspar, "SMI++ - An Object Oriented Framework for Designing Distributed Control Systems", IEEE Trans. Nucl. Sci., Vol 45, Num 4, p. 1946-1950.

[6] S. Lueders, "Computing and Networking Infrastructure for Controls", ICALEPCS07, Knoxville, USA 2007.