The Error and Alarm system for the data acquisition (DAQ) and Distributed Monitored System (CMS) at CERN is successfully used for the physics runs at Large Hadron Collider (LHC) and serves as a distributed CMS online system using a high maximum trigger rate of 100 kHz. The Error and Alarm system has to provide the facilities to retrieve, process, store and display errors and alarms occurred during the operation of the CMS experiment. The DAQ is composed by few thousands of hosts [15] and (20000) interconnection applications which can be monitored in near real-time. One of the critical success factors for an error and alarm management is scalability, several dimensions of scaling requirements, [19][20] (e.g. geographical) and have been considered to cover all aspects of the system design. The main purpose of the CMS alarm and error system is to alert the operator when an abnormal operation occurs, but also to keep history of all past problems for post mortem analysis. The presented design is based on four post mortem concepts: only notify important conditions, notify in time, respond, and provide guidance. The suggested infrastructure fits these needs by providing a set of expandable and reusable solutions allowing use of the alarming system for development, test and operation scenarios.

**ARCHITECTURE AND DESIGN**

The architecture is based on the XDAQ Monitoring & Alarming Service (XMAS) [9], which provides several plug-ins specialized for specific tasks and virtual systems to support and implement a fully scalable distributed monitoring and alarming system. As shown in Figure 1, the system builds upon a scalable publisher-subscriber [10] service consisting of a pool of eventing applications orchestrated by a load balancer application (broker). When a DAQ application detects an abnormal state, it publishes a message to the central DAQ team and a central alarm system. To achieve this, a common database (broker). Oracle was selected to store all the alarms and errors issued by the CMS sub-detectors. The XDAQ management system allows users to define the rules for the error and alarm behavior, and to configure the error and alarm processing in the CMS data acquisition system.

**REFERENCES**

13. Oracle TimesTen In-Memory Database.
14. Oracle Inc.
16. Spotlight2g Cittolin et al 2007. The CMS operators use the mortem display. The CMS operators use the mortem display.
17. Figure 1. In colour the element involved in the reporting, recording and display of errors and alarms.
18. Figure 2. Error and Alarm interaction diagram.
19. BENCHMARKS

The maximum performance achieved for the CMS error and alarm system depends on the database capabilities. SpotlightOCCI has been used to test the error and alarm application at approximately 600Hz (insertions per second). Using a TimesTen memory cache, spotlightTT was able to perform approximately at 2kHz. Spotlight2g, for a core poisoned approximately 1kHz insertions per second.

**PERSISTENCY**

The spotlight application is responsible for the storage and retrieval of error and alarm data. Three spotlight applications have been implemented to support different requirements: Spotlight2g utilized a database called SQLite to store the exceptions. SQLite is a process library that implements a self-contained, server-less, zero-configuration and transactional SQL database engine. SpotlightOCCI, making use of Oracle’s OCCI API, works to store this exception data in an Oracle database. SpotlightTT, utilized in a memory database called XaaS TimesTen [15]. This database is designed for low latency, high-volume data, event and transaction management.

**CONTACTS**

Organization: CERN
Email: lucien.chavignol@cern.ch
Phone: +41 22 767 1615

Name: Andrea Petrucci
Organization: CERN
Email: Andrea.Petrucci@cern.ch
Phone: +41 22 767 0808

**INTRODUCTION**

The Compact Muon Solenoid (CMS) [1] is a general-purpose particle detector at the Large Hadron Collider (LHC) [2] at CERN in Geneva, Switzerland. The CMS Data Acquisition (DAQ) [3] is responsible to build and filter events from the detector's raw data by using a maximum trigger rate of 100 kHz. The Error and Alarm system has to provide the facilities to retrieve, process, store and display errors and alarms occurred during the operation of the CMS experiment. The DAQ is composed by few thousands of hosts [4] and (20000) interconnection applications which can be monitored in near real-time. One of the critical success factors for an error and alarm management is scalability, several dimensions of scaling requirements, [5][6] (e.g. geographical) and have been considered to cover all aspects of the system design. The main purpose of the CMS alarm and error system is to alert the operator when an abnormal operation occurs, but also to keep history of all past problems for post mortem analysis. The presented design is based on four post mortem concepts: only notify important conditions, notify in time, respond, and provide guidance. The suggested infrastructure fits these needs by providing a set of expandable and reusable solutions allowing use of the alarming system for development, test and operation scenarios.

**VISUALIZATION**

The architecture is based on the XDAQ Monitoring & Alarming Service (XMAS) [9], which provides several plug-ins specialized for specific tasks and virtual systems to support and implement a fully scalable distributed monitoring and alarming system. As shown in Figure 1, the system builds upon a scalable publisher-subscriber [10] service consisting of a pool of eventing applications orchestrated by a load balancer application (broker). When a DAQ application detects an abnormal state, it publishes a message to the central DAQ team and a central alarm system. To achieve this, a common database (broker). Oracle was selected to store all the alarms and errors issued by the CMS sub-detectors. The XDAQ management system allows users to define the rules for the error and alarm behavior, and to configure the error and alarm processing in the CMS data acquisition system.

**REFERENCES**

13. Oracle TimesTen In-Memory Database.
14. Oracle Inc.
16. Spotlight2g Cittolin et al 2007. The CMS operators use the mortem display. The CMS operators use the mortem display.