The Monitoring, Logging, and Alarm systems for the Cherenkov Telescope Array

Alessandro Costa[1,4], Kevin Munari[1,4], Federico Incardona[1,4], Pietro Bruno[1,4], Stefano Germani[2,4], Alessandro Grillo[1,4], Igor Oja[3,4],Eva Sciacca[1,4], Ugo Becciani[1], Mario Raciti[1]

for the CTA Consortium

[1]INAF, Osservatorio Astrofisico di Catania, Via S Sofia 78, I-95123 Catania, ITALY,
[2]Università di Perugia, Dipartimento di Fisica e Geologia, IT
[3]CTAO gGmbH
[4]For The CTA Consortium see www.cta-observatory.org

ABSTRACT

We present the current development of the Monitoring, Logging and Alarm subsystems in the framework of the Array Control and Data Acquisition System (ACADA) for the Cherenkov Telescope Array (CTA). The Monitoring System (MON) is the subsystem responsible for monitoring and logging the overall array (at each of the CTA sites) through the acquisition of monitoring and logging information from the array elements. The MON allows us to perform a systematic approach to fault detection and diagnosis supporting corrective and predictive maintenance to minimize the downtime of the system. We present a unified tool for monitoring data items from the telescopes and other devices deployed at the CTA array sites. Data are immediately available for the operator interface and quick-look quality checks and stored for later detailed inspection.

The Array Alarm System (AAS) is the subsystem that provides the service that gathers, filters, exposes, and persists alarms raised by both the ACADA processes and the array elements supervised by the ACADA system. It collects alarms from the telescopes, the array calibration, the environmental monitoring instruments and the ACADA systems. The AAS sub-system also creates new alarms based on the analysis and correlation of the system software logs and the status of the system hardware providing the filter mechanisms for all the alarms. Data from the alarm system are then sent to the operator via the human-machine interface.

INTRODUCTION

The Cherenkov Telescope Array (CTA) will be the largest and most advanced ground-based facility for detection of very-high-energy electromagnetic radiation, from 20 GeV to 300 TeV. When entering the atmosphere, this radiation generates secondary charged particle cascades that can be detected directly or, as in the case of CTA, through the Cherenkov radiation they emit. CTA will be composed of tens of telescopes deployed at North and South Hemispheres.
to achieve full-sky coverage, and an arcminute angular resolution at energies higher than 1 TeV. Typical phenomena that can be investigated include supernovae, supernova remnants, pulsars and pulsar wind nebulae, binary stellar systems, interacting stellar winds, various types of active galaxies, gamma-ray bursts, and gravitational wave transients. By means of its observation, CTA is expected to shed light on some unresolved astrophysics questions such as the role of relativistic cosmic particles on star formation and galaxy evolution, the physics in the proximity of neutron stars and black holes, or the nature of the dark matter. Together with the scientific data produced by CTA, a big volume of housekeeping and auxiliary data coming from weather stations, instrumental sensors, logging files, etc., must be collected as well. In order to ingest the whole amount of data coming from tens of telescopes, a complex software architecture is required that must be able to face such a cutting-edge technological challenge.

Architecture

The *Monitoring and Logging* subsystems (**MON**) provide services for monitoring data items from the Telescopes and other devices deployed at the CTA array sites and making those data immediately available for the operator interface and for quick-look quality checks, as well as to store them for later detailed inspection. The monitoring system works continuously to record any monitoring data made available by Array Elements, which also includes the data points required for engineering purposes.

Monitoring and Logging Supervisor: it receives startup and shut-down commands by the ACADA RM (Resource Manager) and passes them to the MON systems. It provides its status information to the RM. The Resource Manager is the ACADA subsystem that supervises the monitoring logging and alarm subsystems.

Environmental Conditions Inspector: A component that accumulates the data from the central calibration devices to produce indicators for the status of the environment. Most of the environmental monitoring data is received directly from the monitoring system. The component processes the monitoring data, puts associated data elements together, and determines and stores the status of the environment.

Monitoring Value Inspector: A component that provides the capability to re-sample the monitoring information coming in the form of irregular and unevenly spaced time series data to a consistent and regular frequency. The component processes the monitoring events and raises alarms if data are above the threshold for a predefined period of time.

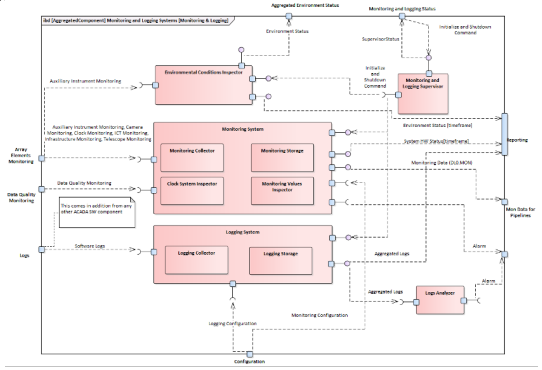

Fig. 1: Monitoring and Logging Subsystem Architecture

The *Logging System* is composed of the following main building blocks:

Logging System: Gets logging information from relevant software components and stores it. This logging comes in three flavours: software logs provided by elements using the control framework, software logs of the observation scripts, software logs produced by low-level firmware, that require reformatting to adapt to the rest of the logs.

Logs Analyzer: A tool to analyse logging data information to trigger further alarms, as well as warnings for the technical crew.

The Array Alarm System (**AAS)** is composed of the following main building blocks:

Alarm Collector: Provides the services to collect any alarm raised by the Array Elements or ACADA components.

Alarm Filter: Provides means to filter, merge and reduce alarms according to defined rules.

Alarm Rules Database: A database defining the alarm reduction rules for the Alarm Filter.

Alarm Storage: Local repository to store alarms and reactions to alarm history.

AAS Supervisor: Manager component for the AAS, connected with to the supervision tree provided by RM.

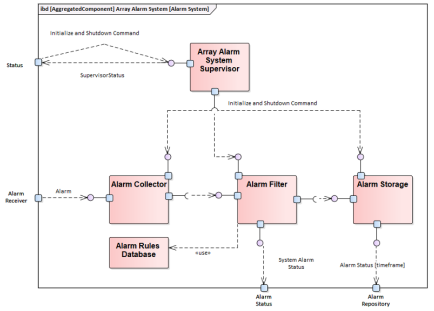

Fig. 2: Array Alarm System Architecture

Technologies

MON and AAS systems are developed in JAVA and they are integrated with the ALMA Common Software (ACS), an open-source framework on which the software operating the ALMA observatory is based on.

More specifically, MON can access and monitor ACS and OPC-UA data sources. To this aim, MON makes use of Eclipse Milo SDK, which provides a pure-Java, open-source implementation of the OPC-UA 1.03 client and server specifications. To exchange the acquired data among the heterogeneous ACADA subsystems, we opted for Apache Avro, a data serialization framework that uses JSON for defining schemas the information exchanged must be compliant with. We make use of Apache Kafka, a distributed event streaming platform designed to handle data streams from multiple sources and deliver them to multiple consumers.

Besides, to forward and centralize logs generated by ACADA, we use a set of distributed lightweight shippers based on Elastic Filebeat. Those log events are ingested, filtered and manipulated by a centralized log aggregator based on Elastic Logstash, which acts as a data processing pipeline that, in the end, sends them to Apache Kafka.

We opted for Apache Cassandra as our database management system (DBMS), which is specifically designed to handle large amounts of data.

We make use of the Docker platform to easily distribute, replicate and scale our deployment environment, packaging the technologies described above in containers.

Aknowledgements

This work was conducted in the context of the CTA Consortium. We gratefully acknowledge financial support from the agencies and organizations listed here: http://www.cta-observatory.org/consortium_acknowledgments