Big Data Architectures for Logging and Monitoring Large Scale Telescope Arrays

A. Costa[1], E. Sciacca[1], G. Tosti[2], J. Schwarz[3], P. Bruno[1], A. Calanducci[1], A. Grillo[1], F. Vitello[1], V. Conforti[4], F. Gianotti[4]
U. Becciani[1], S. Riggi[1]

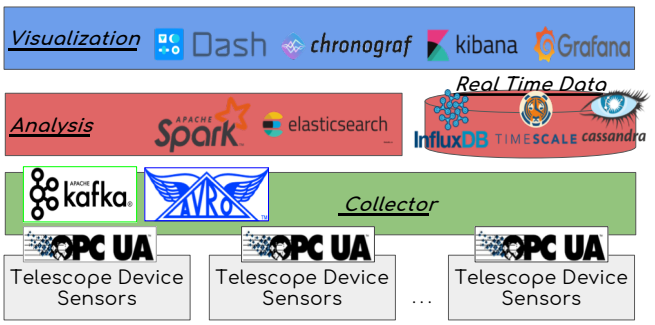
[1]Astrophysical Observatory of Catania, INAF, IT [2], University of Perugia, Department of Physics and Geology, IT
[3]Astrophysical Observatory of Brera, INAF, IT, [4]Astrophysical Observatory of Bologna, INAF, IT

alessandro.costa@inaf.it

Large volumes of **technical and logging data** result from the operation of **large scale astrophysical infrastructure**.
In the last few years several **"Big Data" technologies** have been developed to deal with a huge amount of data, e.g. in the **Internet of Things (IoT)** framework.
We are comparing different stacks of Big Data/IoT architectures including **high performance distributed messaging systems, time series databases, streaming systems, interactive data visualization**.
The main aim is to classify these technologies based on a set of use cases, with the objective to have a system that can be updated, maintained and customized with a minimal programming effort.

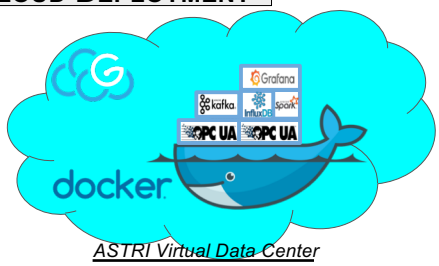
- **ASTRI (Astrofisica con Specchi a Tecnologia Replicante Italiana)** started as a MIUR flag project approved in 2010 to support the development of technologies within the **Cherenkov Telescope Array**.
- The first result of the ASTRI project was the construction of a **prototype telescope** now installed at **Serra La Nave (INAF-Catania)**.
- The ASTRI prototype is a telescope for Cherenkov astronomy and adopts the **Schwarzschild-Couder optical configuration** that uses **two mirrors** making the telescope more compact and reducing the image size on the focal plane.
- The **next phase** of the project, currently underway, is the construction of a series of **9 units of ASTRI telescopes** (named **ASTRI Mini-Array**)

SW Stack



Cloud Deployment

SW Stack deployed on the **GARR Cloud Platform** within a specific **virtual data center** using a **container based virtualization**.
The Cloud infrastructure provides the ability to **simulate a large number of sources** producing IoT data and scale easily the **various modules** according to the needed workload.
Testing sw architecture for an **entire array of telescope**, using any number of modules, each of them running as a separate docker **container**.



ASTRI Virtual Data Center

Future Deevelopments

- Scalability tests and evaluation of performances simulating different data sources on GARR Cloud.
- Integration of Machine Learning algorithms to perform **anomaly detection** and **failure prediction**.
  - Collection of a representative dataset for training
  - Spark implementation

Poster Presenter

Alessandro Costa
INAF OACT Italy

ASTRI Telescopes

The ASTRI Telescopes' main **sub-systems** are:
- Sky Quality Monitor (SQM)
- Telescope Camera (TC)
- Telescope Control Unit (TCU)
- Technical Health Control Unit (THCU)
- Pointing Monitor Camera (PMC)

Some of the mini-array **auxiliary devices** are:
- Weather Station (WS)
- All Sky Camera (ASC)
- UV-SiPM (UVM)

The interface to the hardware devices is represented by **OPC-UA protocol**. This allows to decouple high-level control software from the specific hardware device used and from proprietary communication protocols.

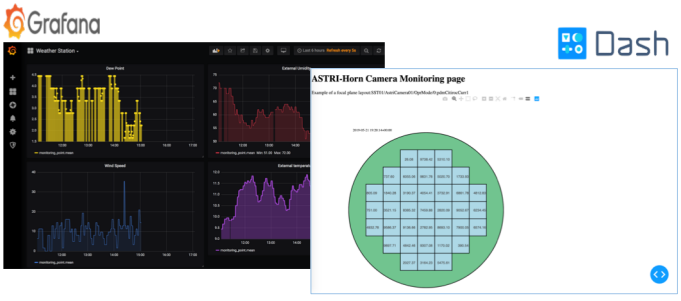
Data Workflow

- **Devices parameters** are sent through **OPC-UA protocol** and described by an **Interface Control Document (ICD)**
- The ICD comprises, in form of tables, for each control or **monitoring point**, a **complete description** of the information required, e.g. data type, OPC-UA node and connected alarms.
- In our application, the **IoT data producer** is an **OPC-UA simulator application** for connected devices.
- Data are collected using an **OPC-UA client** and sent through an **Apache Kafka Producer** over a given **topic** serialized using an **Apache Avro** schema:

```
{
  "namespace": "avro.example",
  "type": "record",
  "name": "MonitoringPoint",
  "fields": [
    {"name": "id", "type": "string"},
    {"name": "env_id", "type": "string"},
    {"name": "serial_no", "type": "string"},
    {"name": "date", "type": "long"},
    {"name": "data", "type": { "type": "array", "items": "double"} }
  ]
}
```

```
{
"id": "SST01/WeatherStation01/ws_windspd",
"env_id": "SLN",
"serial_no": "WS-01",
"date": 13203093272171,
"data": [20.921472]
}
```

- A **Kafka Consumer** get the **deserialized** data from the subscribed topic.
- The Spark application analyzes the data stream and insert the analysis results into a non-relational database (optimized for real-time and Big Data applications, such as Cassandra or InfluxDB).
- Finally, monitoring data are visualized through interactive dashboards.



DATA TYPES

- EVT: (event) contains data that changes for every triggered event (e.g. a Air-Shower Event, Array-Level Event, Local Event, Camera Event), with typically a high rate, which may be more than a kilohertz at the R0-DL0 level. For this reason, EVT data may need special storage considerations. Examples include shower images/cubes, shower parameters, calibration coefficients that are measured event-wise, and trigger information. This is typically the highest volume and most complex data stream. (indexed-by: event id)
- MON (monitoring) contains time-series data that are used to monitor the status or quality of hardware, algorithms, or other data products. These typically update periodically during the operation of the array, or during data processing, at a rate typically much slower than EVT data and faster than the length of a typical Observation Block. Examples would be slow-control information like tracking positions, weather monitoring data, or the status or quality-control data of a particular hardware or software component. (indexed-by: timestamp)
  - From
  - - Auxiliary Instruments
  - - Infrastructure Elements
  - - Telescopes
  - - Data Quality
- LOG: from computing systems
- SVC (service) contains data that act as a service to an observation, hardware or software component. Examples include calibration look-up-tables or instrumental-response functions. Generally these are quantities that change rarely. This data must include a validity range or a list of dependent/parent data products. (indexed-by: version, validity-range)