ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA032

EPICS AND OPEN SOURCE DATA ANALYTICS PLATFORMS

C. R. Haskins, CSIRO Astronomy and Space Science (CASS), Epping, Australia

Abstract

SKA scale distributed control and monitoring systems present challenges in hardware sensor monitoring, archiving, hardware fault detection and fault prediction. The size and scale of hardware involved and telescope high availability requirements suggest that machine learning and other automated methods will be required for fault finding and fault prediction of hardware components. Modern tools are needed leveraging open source time series database & data analytic platforms. We describe DiaMoniCA for The Australian SKA Pathfinder Radio Telescope which integrates EPICS, our own monitoring archiver MoniCA, with an open source time series database and web based data visualisation and analytic platform.

INTRODUCTION

The Australia Square Kilometre Array Pathfinder (ASKAP) Radio Telescope has been in operation for several years while dishes and associated digital systems for each of its 36 antennas are commissioned. During this time, the in-house developed MoniCA data archiver platform [1] has been in use. As the ASKAP Telescope Operating System (TOS) uses the Experimental Physics and Industrial Control System (EPICS) framework for monitoring & control, an EPICS Channel Access archiver plugin was developed for use with MoniCA.

The MoniCA Server and MoniCA data visualisation client (MoniCA Client) are used across all CASS Radio Telescopes, providing some continuity to the otherwise very different instruments. However, the MoniCA platform is aging and the scalability requirements of ASKAP and beyond require new solutions. Migration to web-based solutions are seen as essential for ease of deployment and maintainability over the life of ASKAP. By adopting solutions from the wider DevOps community, we can take advantage of the latest technology trends, gain scalability and draw from a much larger pool of open source contributions.

DiaMoniCA is a CASS platform that encompasses all aspects of monitoring, visualisation and data mining for ASKAP and other CASS instruments. It brings together a collection of open source technologies and in house developed modules to integrate off the shelf data archiving & visualisation tools with EPICS monitoring and control.

Figure 1 shows the main components of DiaMoniCA. The platform is centred around a Commercial Open Source time series database and a data visualisation platform, together with in-house developed modules for data ingest and data extraction. Ancillary services such as alerting are provided as standard components of the open source platforms.

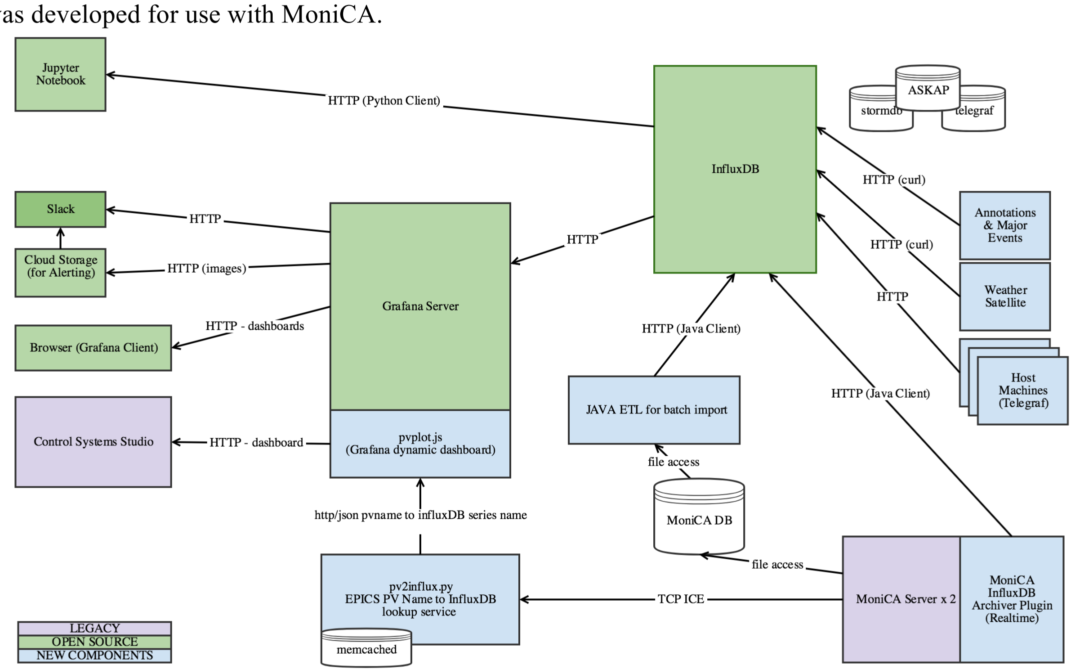

Figure 1: The DiaMoniCA platform in ASKAP.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA032

InfluxDB

InfluxDB is the "I" in InfluxData's TICK stack [2]. It is a highly scalable time series database, designed for high volumes of data at high data rates and as such is a good fit for the requirements of ASKAP and beyond. Data can be ingested into InfluxDB using either Telegraf ("T" in the TICK stack), a client library (as used in MoniCA Server) or directly via the HTTP API.

Grafana

While InfluxData does have its own web based data visualisation front end with Chronograf ("C" in the TICK stack), the Open Source platform Grafana from Grafana Labs [3] comes with InfluxDB support and is widely supported by an active community through development of plugins. It also supports several other data sources, bringing together monitoring data from a variety of source such as Weather Satellite data, Building Management Systems (BMS) and High Performance Computing (HPC) monitoring of the ASKAP imaging pipeline.

MIGRATION STRATEGY

In order to prove out InfluxDB and Grafana as a replacement for MoniCA, we have adopted a staggered approach which first requires running MoniCA and InfluxDB side by side for a period of time. Once all capabilities have been replicated in DiaMoniCA, MoniCA can be decommissioned.

Batch Import of Historical Data

To evaluate the performance of InfluxDB as an archiver for multiple years of data, we enlisted the help of CSIRO's Data61 [4] to provide an Extract, Transform, Load (ETL) process for migrating historical ASKAP data from MoniCA to InfluxDB.

We were able to use this process to refine the InfluxDB schema design to maximise performance while adhering to the InfluxDB schema design guidelines.

From there we were in a good position to examine the performance of InfluxDB and identify it's benefits over the existing propriety solution.

Real Time Ingest

As part of the migration strategy, real-time ingest is achieved through the use of a MoniCA archiver plugin. Traditionally the MoniCA server archives data in a compressed ASCII format with the directory structure forming a fast index into the time series data. Rather than replacing MoniCA up front, we can extend MoniCA with an alternate archiving strategy (Figure 2).

Eventually the ASCII database can be decommissioned and the MoniCA Server can be optimized for InfluxDB as the primary database backend.

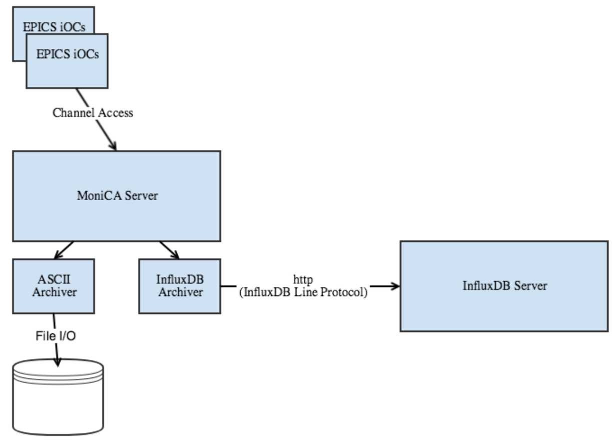

Figure 2: Real-time ingest with MoniCA plugin.

DATA VISUALISATION

Fast, easy to use and extensible data visualisation tools are necessary when examining time series data from a system as complex as ASKAP. While Grafana wasn't identified as a primary goal when starting the evaluation of InfluxDB, it's ease of integration and the quality of the user experience elevated it to a centre piece of the DiaMoniCA platform.

Integration with Control Systems Studio (CSS)

For the initial development of ASKAP TOS, The CSS Data Browser plugin for CSS was extended to allow reading directly from the MoniCA server via an ICE interface. While this solution worked, the performance associated with the MoniCA server lookup was poor when querying several EPICS Process Variables (PV) at a time.

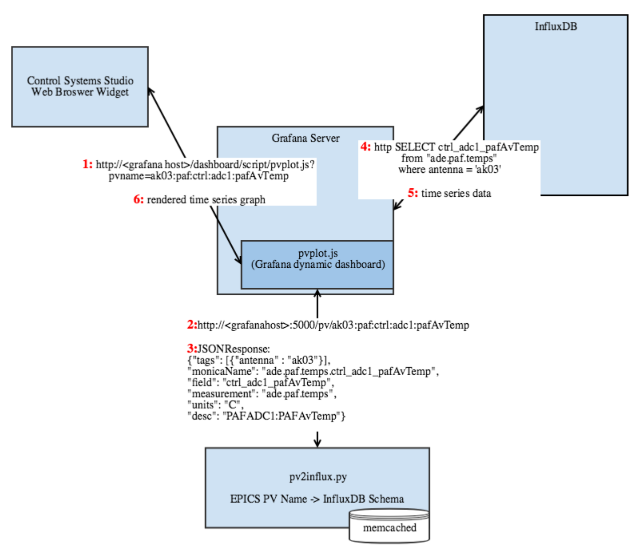

Figure 3: Grafana integration with CSS.

Figure 3 shows an alternate solution using a Grafana dynamic dashboard. A CSS action button launches a web

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-THPHA032

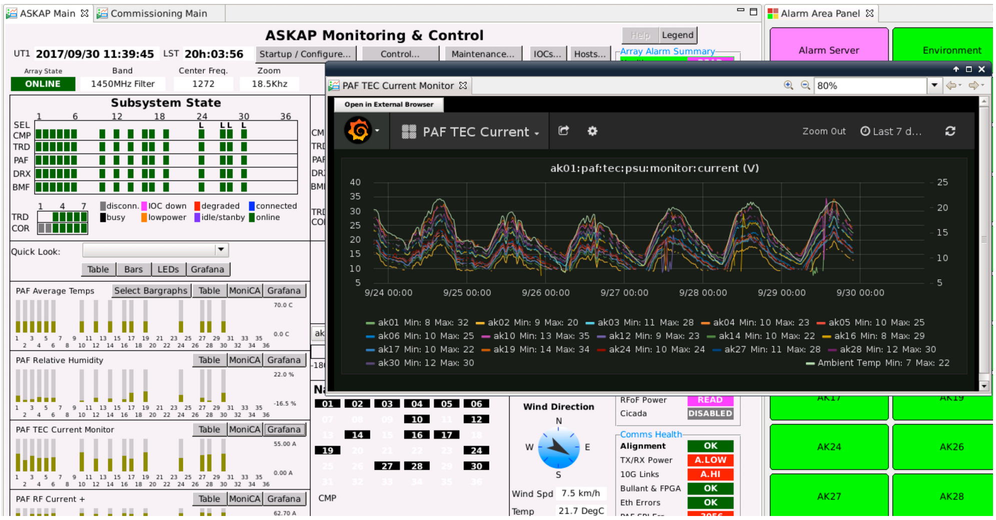

Figure 4: Grafana integration with CSS.

browser widget with the URL containing the EPICS PV name. The PV name is passed to the Grafana dynamic dashboard, pvplot.js which in turn queries a PV name to InfluxDB schema lookup service. The lookup service sends a JSON response containing the InfluxDB measurement name, field name and tags needed for the database SELECT query. Grafana can then query the InfluxDB server for the data associated with the EPICS PV. Additional parameters can be passed to pvplot.js such as time span and plotting options. By using an in-memory cache (memcached) in the lookup service, the whole process from click to rendered plot takes less that 2 seconds when displaying a 7-day history across of multiple PVs.

```
import numpy as np
from influxdb import DataFrameClient
client = DataFrameClient(database='askap', host='akingest01')
```

```
# query air temp for last 30 days
result=client.query("SELECT mean(Temperature) as \"Ambient Temp\" \
FROM \"environment.weather\" WHERE time > now() - 30d GROUP BY time(1h)")
airTempData = result['environment.weather']

# and antenna (PAF) temp
pafData = client.query("SELECT mean(ctrl_adc1_pafAvTemp) as meanPafTemp \
FROM \"ade.paf.temps\" WHERE time > now() - 30d GROUP BY antenna, time(1
h)")
```

```
# plot both together
%matplotlib inline
ax = airTempData.plot()
for key in sorted(pafData.keys(), key = lambda x : x[1][0][1]):
    pafData[key].columns = [key[1][0][1]]
    pafData[key].plot(ax=ax,legend=True,title="PAF Temperatures", figsize=
(12,6))

tmp=ax.legend(loc='upper center', bbox_to_anchor=(0.4, -0.05), ncol=7)
tmp=ax.set_ylabel("Degrees C")
```

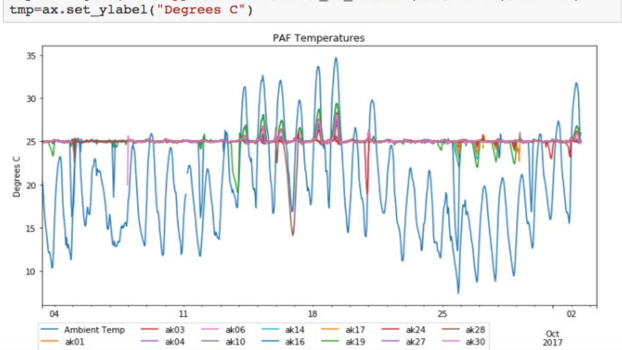

Figure 5: Data exploration with Jupyter Notebooks.

The Grafana approach yielded a solution with a 10-fold improvement in load time compared to our original solution with the CSS DataBrowser connected to a MoniCA data source. It also provides a richer user experience and a consistent interface for the visualisation of time series data (see Figure 4).

INTERACTIVE DATA EXPLORATION

Prior to the adoption of DiaMoniCA, querying and analysing historical data was accomplished through the use of the MoniCA Client time series plotting tool. With data in an accessible format such as InfluxDB, we can now leverage standard solutions such as extracting sensor data into Python Pandas dataframes [5]. Data exploration can be run interactively within a Jupyter notebook as show in Figure 5. This opens up exploration of data using machine learning toolkits.

CONCLUSION

Migrating to an open source data archiving solution from outside of the EPICS domain has enabled ASKAP to adopt the latest technologies in data archiving and visualisation.

The use of standard platforms allows integration with other data sources and we can leverage the services that are provided such as alerting and anomaly detection.

Overall, the economies of scale gained from adopting these solutions has reduced our reliance on niche EPICS solutions and proven our strategy going forward.

REFERENCES

[1] MoniCA, `http://www.narrabri.atnf.csiro.au/monitor/monica`
[2] InfluxData TICK Stack, `http://www.influxdata.com/time-series-platform/`
[3] Grafana, `http://www.grafana.com`
[4] CSIRO Data61, `http://data61.csiro.au`
[5] Pandas, `http://pandas.pydata.org`