ASKAP DiaMoniCA

EPICS and Open Source Data Analytics Platforms

C R Haskins (CASS)

CSIRO ASTRONOMY AND SPACE SCIENCE
www.csiro.au

SKA scale distributed control and monitoring systems present challenges in hardware sensor monitoring, archiving, hardware fault detection and fault prediction. Tools are needed which can archive large amounts of data while ensuring the data is accessible to users. DiaMoniCA for The Australian SKA Pathfinder Radio Telescope integrates EPICS, our own monitoring archiver MoniCA with an open source time series database and web based data visualisation and analytic platforms.

InfluxDB as an EPICS Channel Archiver

The Australia Square Kilometre Array Pathfinder (ASKAP) Radio Telescope has been in early science operation with the in-house developed MoniCA data archiver platform for several years. We are seeking a new web based platform to take ASKAP to full capacity. InfluxDB is a highly scalable open source 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.

Moving Archived Data to InfluxDB

We first moved historical data from MoniCA to InfluxDB using an Extract, Transform, Load (ETL) process developed by CSIRO's Data61 group. This has allowed us to examine the performance of InfluxDB with a large data set from the beginning.

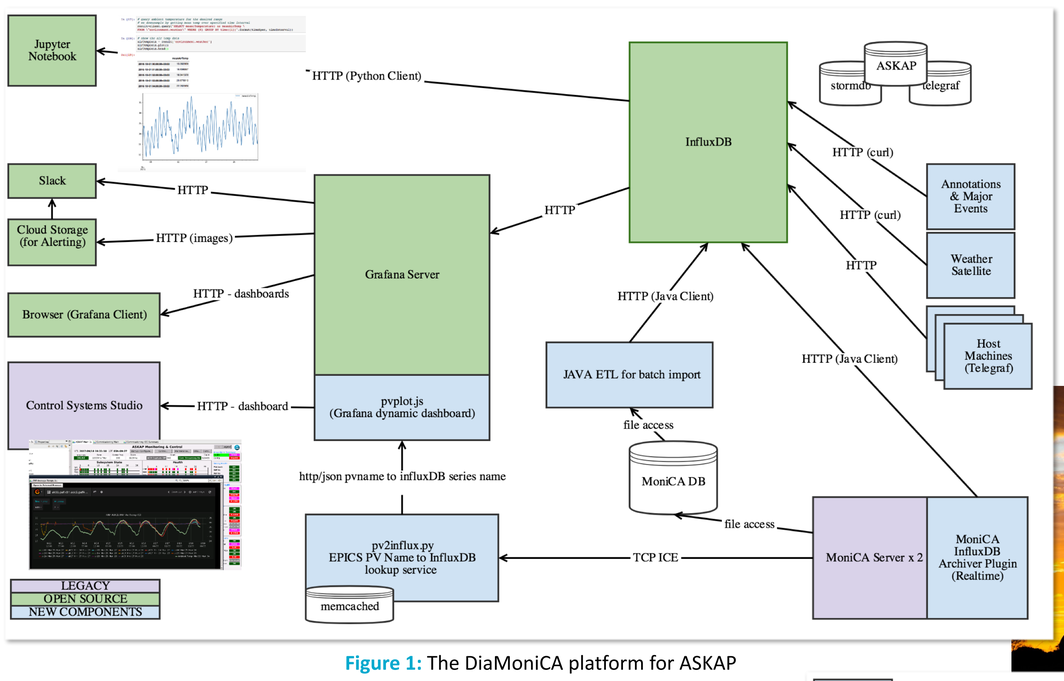

Figure 1: The DiaMoniCA platform for ASKAP

Real-time Ingest

The next step was to create a MoniCA archiver plugin using the InfluxDB Java Client library. During the transition phase, data is archived both to MoniCA and InfluxDB. After a successful trial period, the original MoniCA archive can be decommissioned.

Data Visualisation with Grafana

With EPICS monitoring data in InfluxDB, rich web based open source data visualisation platforms such as Grafana can now be used in place of EPICS only solutions. We become part of a larger user & developer community across a variety of industries.

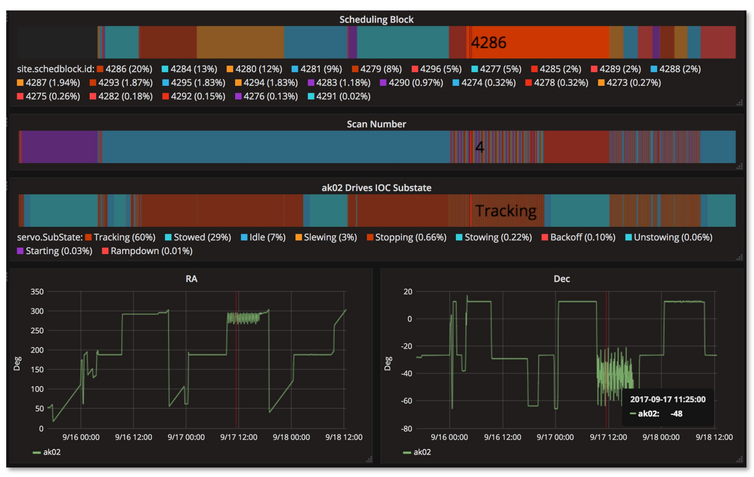

Figure 3: Visualizing EPICS data with Open Source Grafana Plugins

Control Systems Studio (CSS) Integration

- A Grafana dynamic dashboard can be used to render any monitoring point without having to manually create a dashboard view.
- CSS is configured to launch a web browser widget with the PV name embedded in the URL.
- The dashboard, via a http lookup service, builds the correct query expression for InfluxDB and the plot is rendered within CSS.

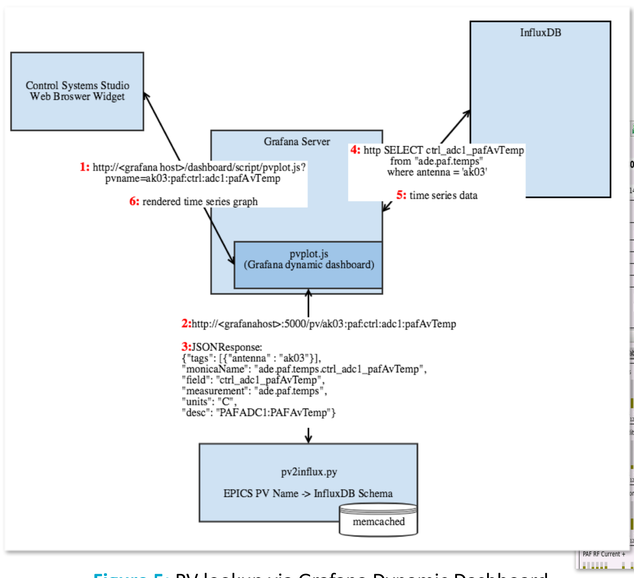

Figure 5: PV lookup via Grafana Dynamic Dashboard

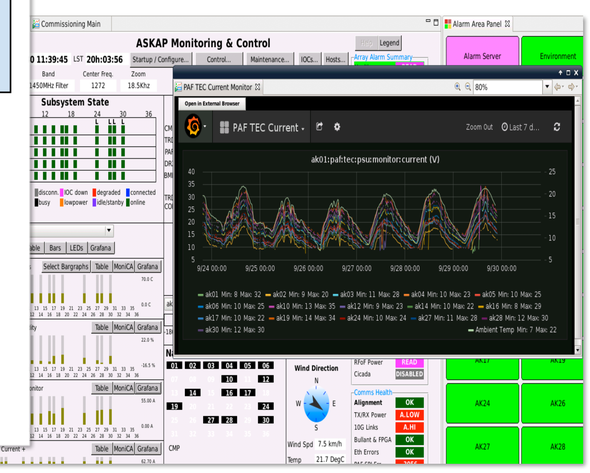

Figure 6: Grafana inside Control Systems Studio

Data Exploration

With the archival data in an open format, standard data analysis tools can be used such as Jupyter Notebooks with Python Pandas.

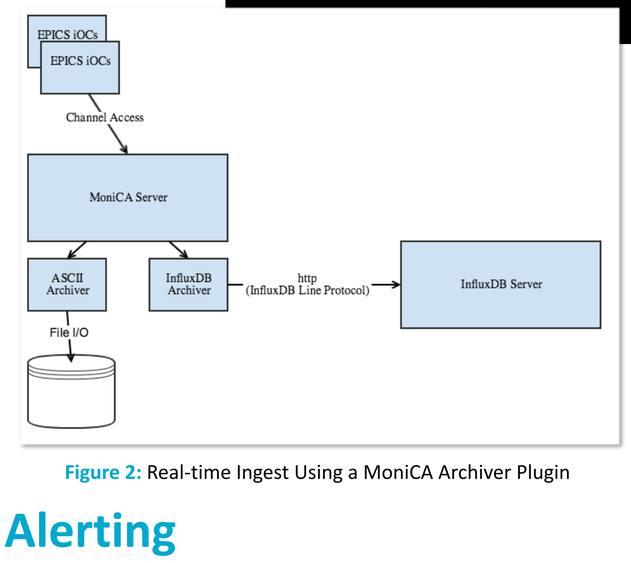

Figure 2: Real-time Ingest Using a MoniCA Archiver Plugin

Alerting

We can also take advantage of services that are built on top of InfluxDB and Grafana such as alerting via Slack.

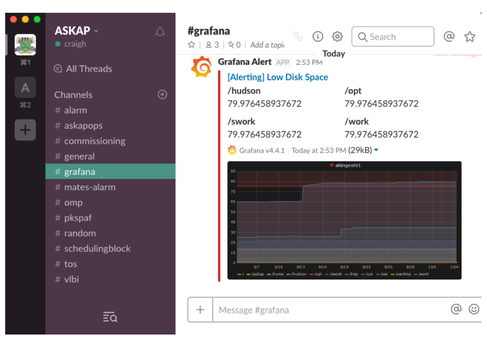

Figure 4: EPICS PV Alert Notifications via Slack

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

# 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")
```

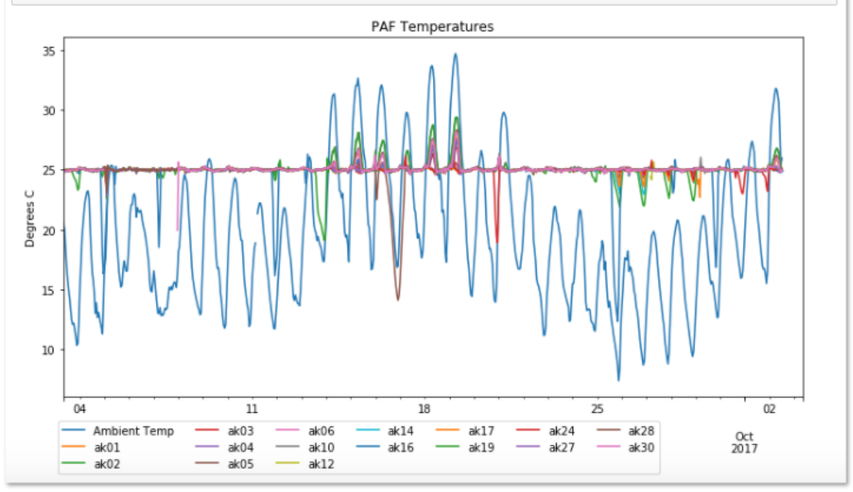

Figure 7: Interactive data exploration of EPICS archival data with Python Pandas

FOR FURTHER INFORMATION

Craig Haskins
e Craig.Haskins@csiro.au
w www.csiro.au/cass