WBS 1.08.02: Monitoring and Alarms

L3 Lead - Cosmin Deaconu

Who am I

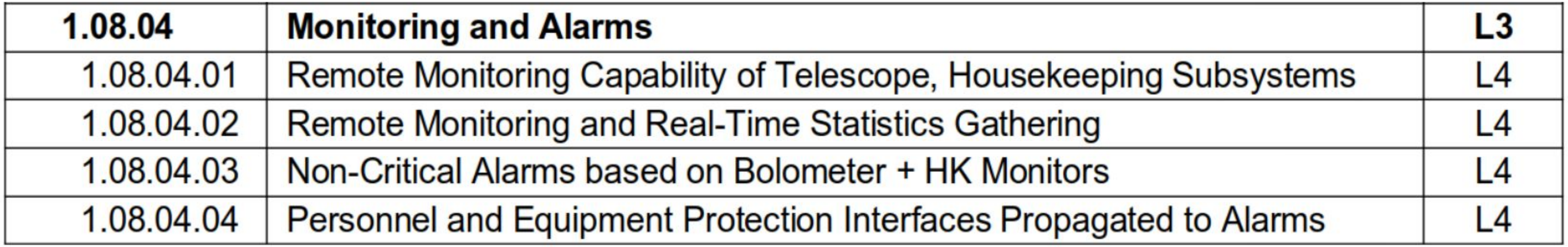
Cosmin Deaconu

1.08.04

- Research Assistant Professor at UChicago
- Primary background in radio-detection of UHE neutrinos (ANITA/PUEO, RNO-G, ARA, BEACON, IceCube Gen2Ratio, RET). PhD in directional dark matter detection
- I have helped to implement monitoring systems for every experiment I've worked on.

Scope of Monitoring and Alarms

- Developing remotely-accessible tools to display current and recent state of housekeeping data for instruments in a facility, to help determine if everything is functioning correctly, and let us know if something is wrong
- We stress, safety alarms are NOT in scope (though, we will propagate them on a best-effort basis)
- Here are the L4s for this L3:



1.08.04	Monitoring and Alarms	L3
1.08.04.01	Remote Monitoring Capability of Telescope, Housekeeping Subsystems	L4
1.08.04.02	Remote Monitoring and Real-Time Statistics Gathering	L4
1.08.04.03	Non-Critical Alarms based on Bolometer + HK Monitors	L4
1.08.04.04	Personnel and Equipment Protection Interfaces Propagated to Alarms	L4

Key Requirements (Design Drivers) for Monitoring and Alarms

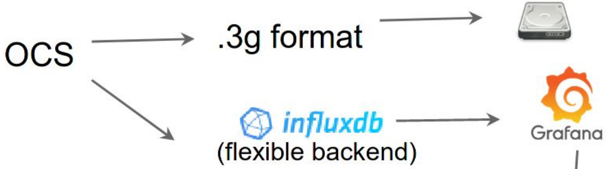
- Near-real-time browser based monitoring of auxiliary (‘housekeeping’) and diagnostic data, including detector metadata.
  - Scalable to an entire site (O (100k fields/sec)) while maintaining O(5 s) latency
  - Flexible and robust enough to be easily adaptable for lab setups and full sites (easy configuration, no hardcoding)
- Decimated views of historical data (>1week)
- Hierarchical alarm system based on housekeeping data
  - Integrating with site-specific systems as necessary
- User-friendly, so users can make plots needed without undue domain-specific knowledge.
- Leverage existing software as much as possible
  - Require open source, for adaptability and longevity

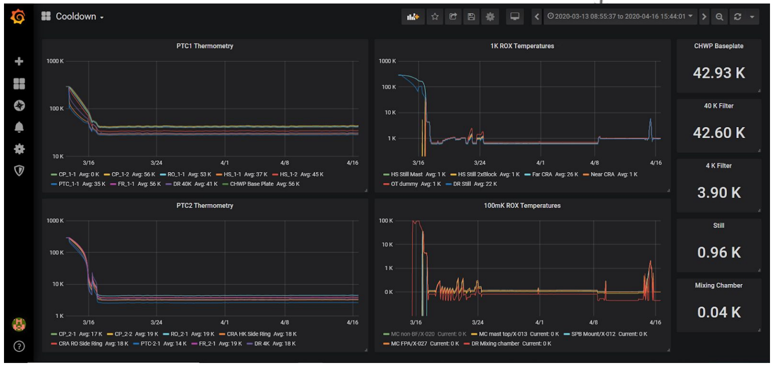
Interfaces

- 1.08.02 (OCS) provides variables to monitor
- 1.08.05 (lab testing) must be a supported use case
- L2 Interfaces:
  - Sites (site specific alarms, integration & commissioning)
  - Data Management (1.9, soft boundary with livelook)

Web-Based Housekeeping Design

- Main web-interface is the popular, open-source **Grafana** package
  - Grafana offers flexible (and extendable!) data provider backends.
  - **InfluxDB** currently the choice for the backend for housekeeping data.
- OCS automatically inserts measured quantities into database.
- Intuitive point-and-click interface for creating dashboards, panels, which can be tailored or reused between setups
  - “Explore” mode for on-the-fly new plots
  - Usable even on phones!
- In heavy use in SO, being prototyped at Pole, also used by other experiments (e.g. RNO-G)

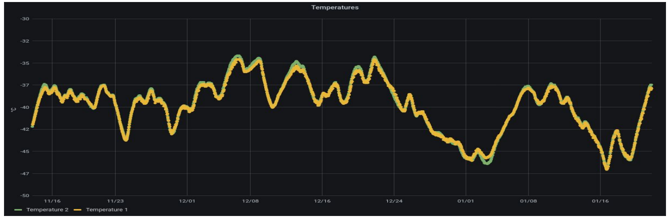




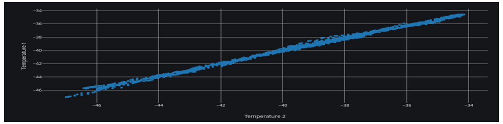
A grafana dashboard, in a web browser

Summaries, Decimation, Correlation

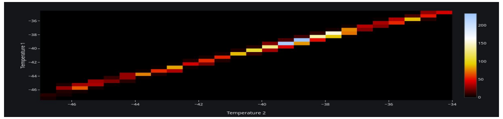
- Many summary variables can be created directly as database queries
  - Time-series databases provide time-series aggregate support (average, min/max, etc.)
  - Summary variables can be displayed on Grafana dashboards (and histograms too!)
- Time-series databases provide support for automatic historical decimation through continuous queries and retention policies
  - Can keep highly-downsampled versions for historical views so monitoring system storage doesn’t grow without bound (and user’s don’t crash their browsers when they try to display a year’s worth of data).
- Grafana plugin system allows creating new plot types, calculations, displays
  - E.g. plotting time-series variables against each other, interpolating if not on same time base



(boring example, two temperature sensors)

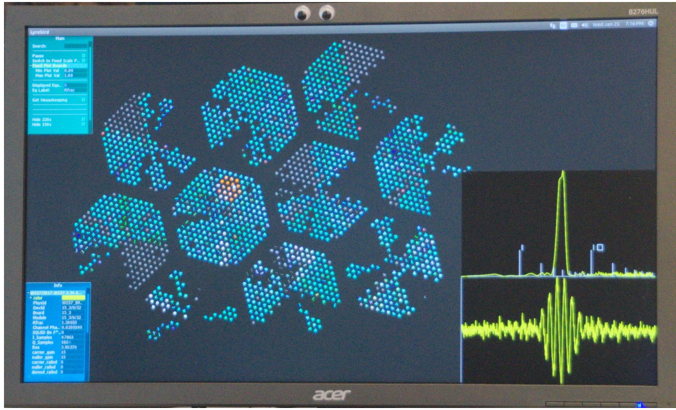


or

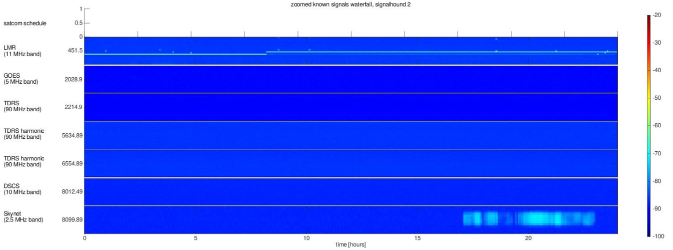


Integrating Non-timestream data

- Not everything is a housekeeping time stream.
- Examples:
  - Detector status: SPT-3G, PB2 and SO use native OpenGL tool (Lyrebird)
    - Used for Live testing, integration, QA
    - Remote monitoring currently via VNC
  - RFI Monitoring: RFI monitors exist at Pole and will in Chile as separate tools
- Design work is ongoing on how to monitor / plot these types of data in integrated way with grafana
  - Prototype binary(.3g)-to-json server, to provide non-timestream data to panel plugins written in web technologies (JavaScript/TypeScript).
  - Waterfall plot panel plugin planned for RFI
  - Potentially webgl version of Lyrebird too (needs feasibility study). Or can punt and use vnc view.



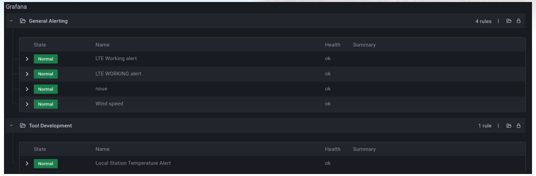
Lyrebird in action



South Pole RFI Monitor

Alarms

- Alarms system with query OCS database and determine if parameters are out of range.
  - All alarm states written to database.
- Alarm displayed and propagated according to severity
  - Lowest severity might be a slack message, escalating to e-mail, and eventually SMS or phone call
  - Will eventually integrate with local alarm systems at sites, e.g. VOIP at Pole
- While we will propagate safety alarms collected by OCS, this is NOT a safety system
  - Required uptime will not be guaranteed.
- Current plan is to use **built-in grafana** alerting system (alarms based on displays), with necessary integrations for alerting methods.
  - This is different from before! Grafana alerting has improved faster than alternative, and this simplest showing alert states on dashboards.



Grafana

General Alerting — 4 rules

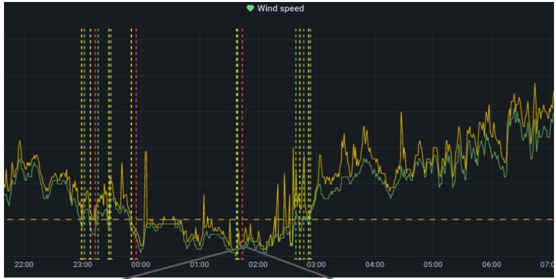
State	Name	Health	Summary
Normal	LTE Working alert	ok	
Normal	LTE WORKING alert	ok	
Normal	noue	ok	
Normal	Wind speed	ok	

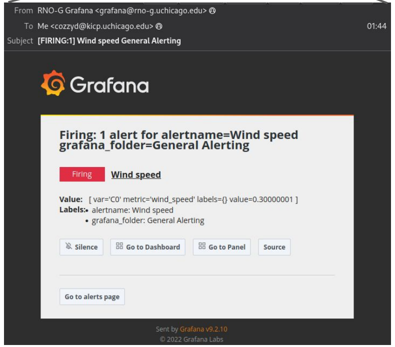
Tool Development — 1 rule

State	Name	Health	Summary
Normal	Local Station Temperature Alert	ok	

(*above*) Grafana current alert states display

(*below*) Alert states displayed on panel (green = ok, yellow = pending, red = firing) and an associated e-mail alert





From RNO-G Grafana <grafana@rno-g.uchicago.edu>
To Me <cozzyd@kicp.uchicago.edu> 01:44
Subject [FIRING:1] Wind speed General Alerting

Grafana

Firing: 1 alert for alertname=Wind speed grafana_folder=General Alerting

Firing **Wind speed**

Value: [var='C0' metric='wind_speed' labels={} value=0.30000001]
Labels:
- alertname: Wind speed
- grafana_folder: General Alerting

Sent by Grafana v9.2.10
© 2022 Grafana Labs

Next steps to CD-1

- Determine what monitoring features users want, and implement if necessary!
- Improved integration of detector live monitoring and other non-timestream data with housekeeping
- Verify scalability of InfluxDB, consider other backends if necessary
- Outline a security policy and an external dependency versioning policy

Summary

- Building blocks in place for powerful, flexible suite of monitoring and alarms tools
  - Grafana as main housekeeping viewing interface. Ideally everything integrated, but perhaps some specialized tools for detectors if necessary
  - InfluxDB as baseline time series database
  - Alarms from querying time series database, integrated in Grafana
- We have time to add features users and want and need to monitor S-4.