Monitoring the readout temperature in the ICARUS TPC

2023 Summer Students Italian Program

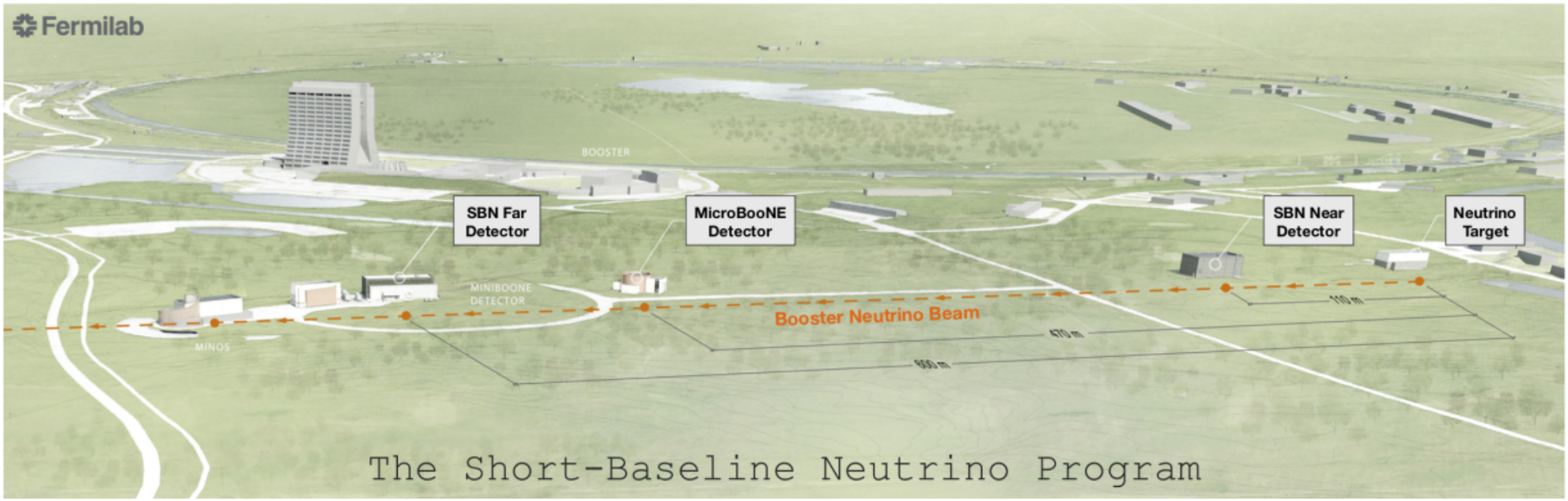
Final Presentation

09/27/2023

Supervisors: Filippo Varanini
Geoff Savage

Student: Giovanni Zago

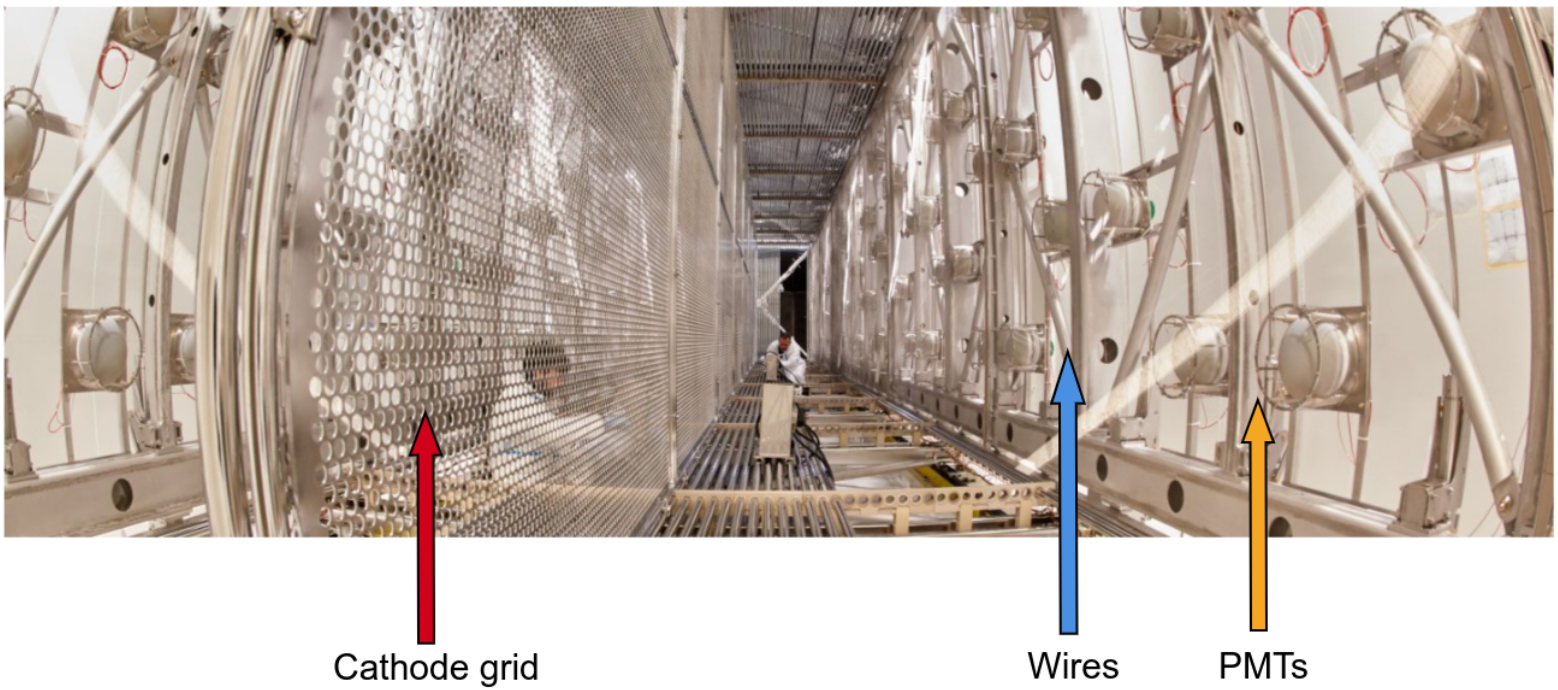
ICARUS and the SBN program at Fermilab



- ICARUS is one of the three experiments that constitute the **Short Baseline Neutrino** program at Fermilab
- The goal of the SBN program is studying the **possible existence of the sterile neutrino** exploiting **two functionally identical detectors** (LAr-TPCs) sitting along the same neutrino beam (Booster Neutrino Beam)

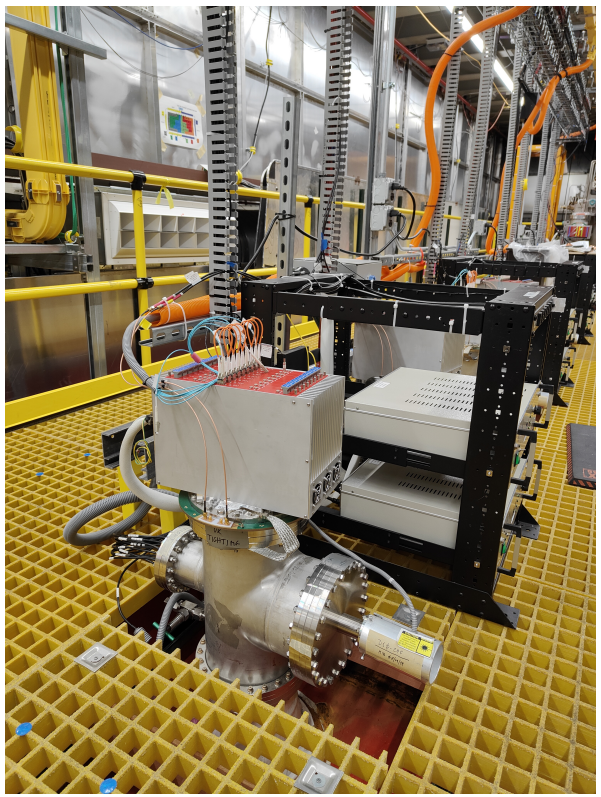
The wires of the LAr-TPC detector

- Being a Time Projection Chamber (TPC) detector, ICARUS exploits a **series of three parallel wire planes** to measure and collect the ionization charge coming from an event inside the active volume
- There's a total of **53,248 wires**, characterized by a diameter of 100 μm, a length up to 9 m and a pitch of 3 mm
- The analog front-end, the ADC and an external optical transmission link are all embedded into **A2795 boards**, arranged in crates located on top of the cryostat
- Each A2795 board is provided with **two temperature sensors**: one is located near the inlet for the fresh air, while the second is located in proximity of the most power-consuming component



Why monitoring TPC board temperatures?

- The board's temperature is key when dealing with the **gain/noise ratio** of the signal coming from the wires. In particular, the charge preamplifiers are the most sensitive to the temperature.
- The boards are designed by the manufacturer to work in the **range (0 - 40) °C** and while the experiment is running the temperature is not that far below the upper boundary.
- Last year (May - July 2022) problems related to board temperatures caused significant losses in terms of run time. Updates have been made in order to improve the airflow in the mezzanine but issues may reoccur.



Goals of the work

At the beginning of my internship the goals were:

- Access old board temperature data in order to perform a complete **temperature characterization**
- **Develop an alerting system** on Grafana in order to monitor the board temperatures

As the work progressed, some **aspects/issues emerged**:

- Grafana data source, i.e. the Carbon database in the *Whisper* format, is a database intended for time series visualization and the Graphite APIs are not suited for accessing the data easily with common tools (Python scripts, Jupyter Notebooks, etc.)
- Moreover, following the Grafana update occurred last month most of the data stored inside the Carbon db became inaccessible
- Independently on the data retention settings, the data fed to the Carbon db is subject to granularity loss as times goes by and this is not good if a high-level analysis has to be performed

Presentation overview

1. **Discussion on Grafana:** alerts, dashboards and panels
2. Solutions for **middle-long term temperature monitoring**

Presentation overview

1. **Discussion on Grafana:** alerts, dashboards and panels
2. Solutions for **middle-long term temperature monitoring**

Grafana alert rules - 1

There are **9 alert rules** currently set up on Grafana:

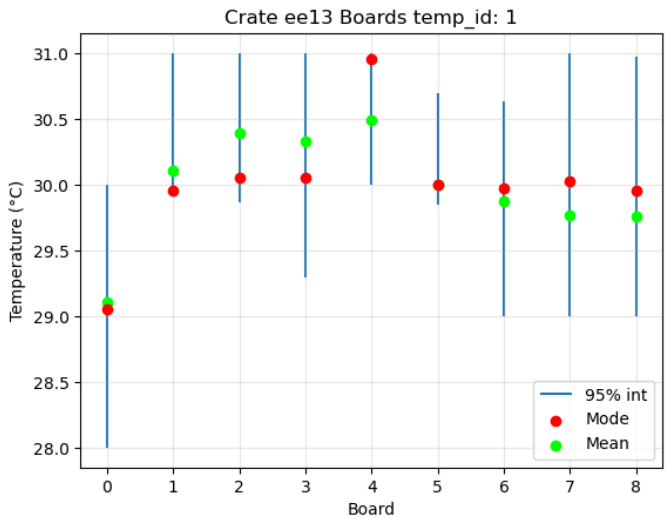
- **8** of them **are temperature-monitoring-related** rules and correspond to the 8 top crates (flanges 1 and 20 for each of the four rows)
- There's **a single NoData alert** that notifies when one the EE01T boards stops sending data

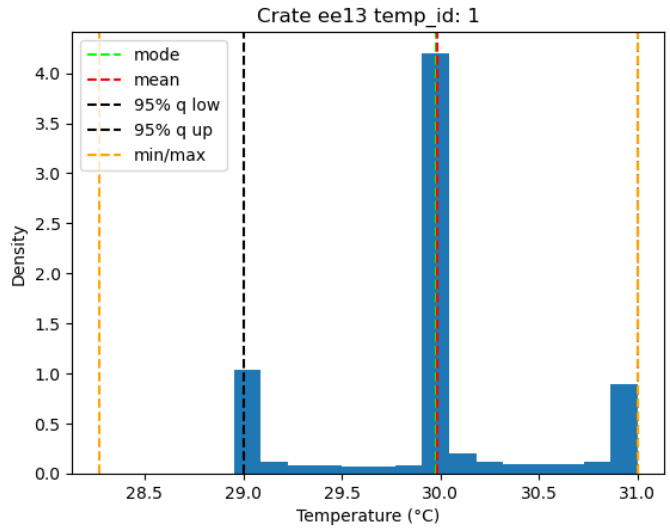
The underlying reasons are, respectively:

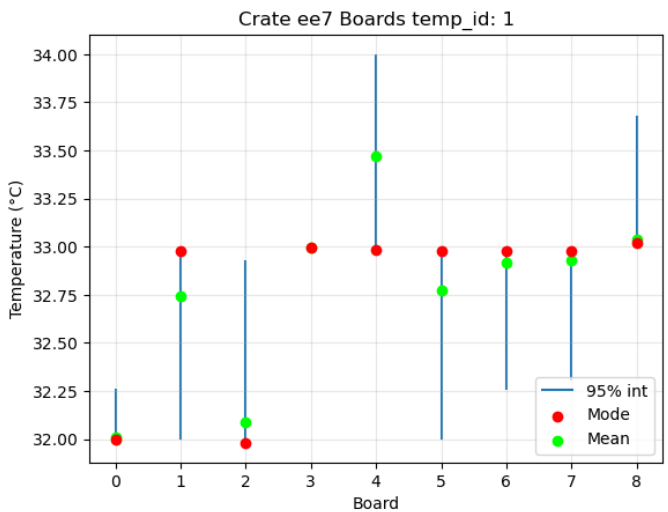
- While characterizing the temperature it was possible to notice that the **crates** located **at the edges** of each row reach **significantly higher maximum temperatures** than the ones at the center of the row
- Regarding the NoData alert, just a single one is needed since, if the run crashes, then every component would display the same NoData status

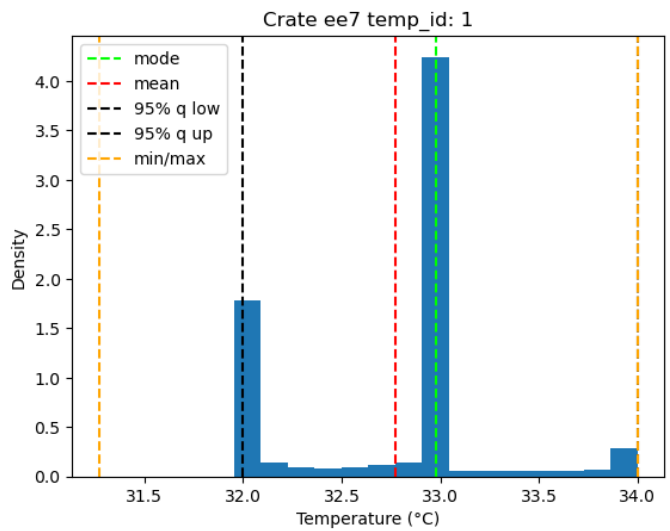
Crate temperature distribution - Run 10235

- Board temperatures **do not share the** same **distribution**
- This lets us think that it is reasonable to **setup alarms on a crate level**

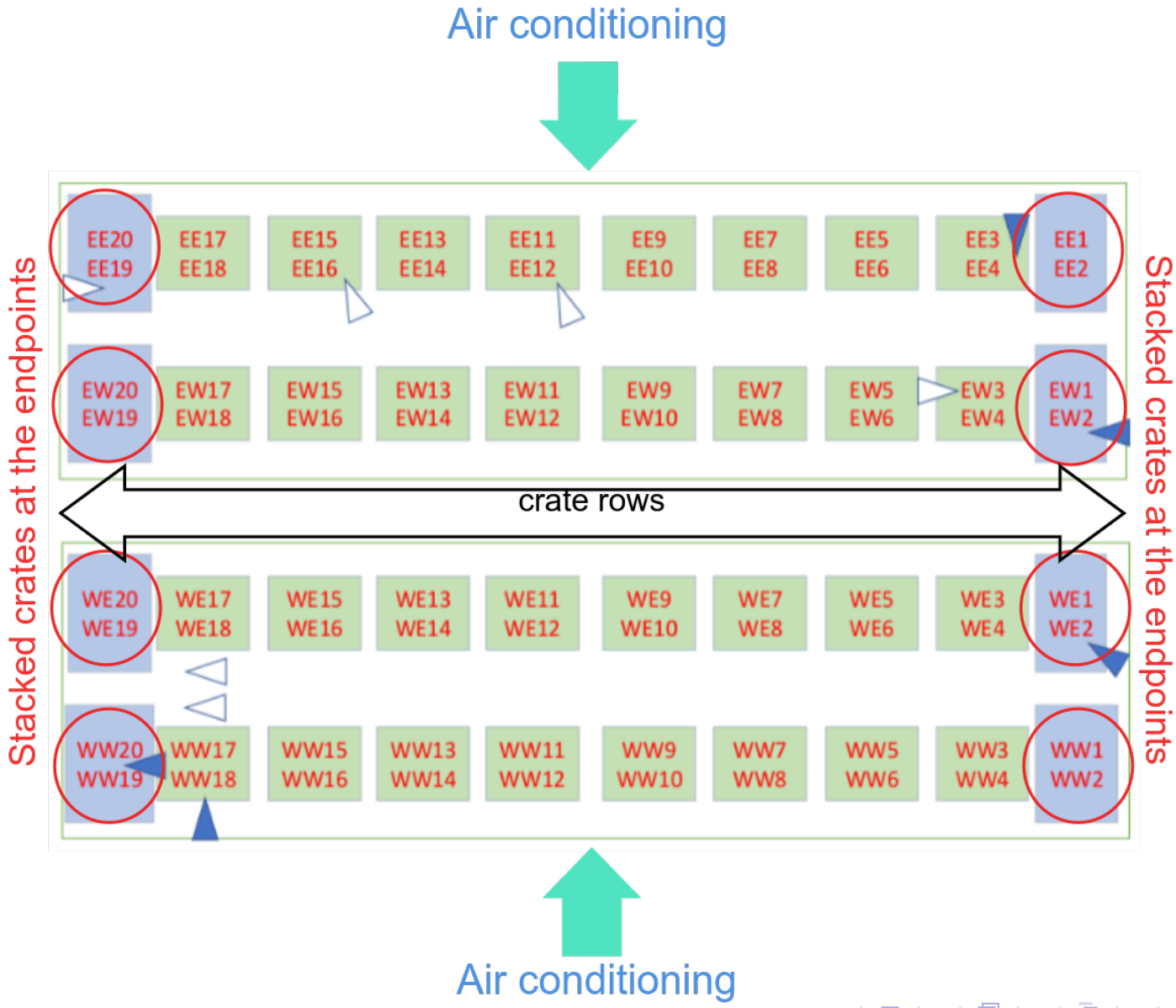






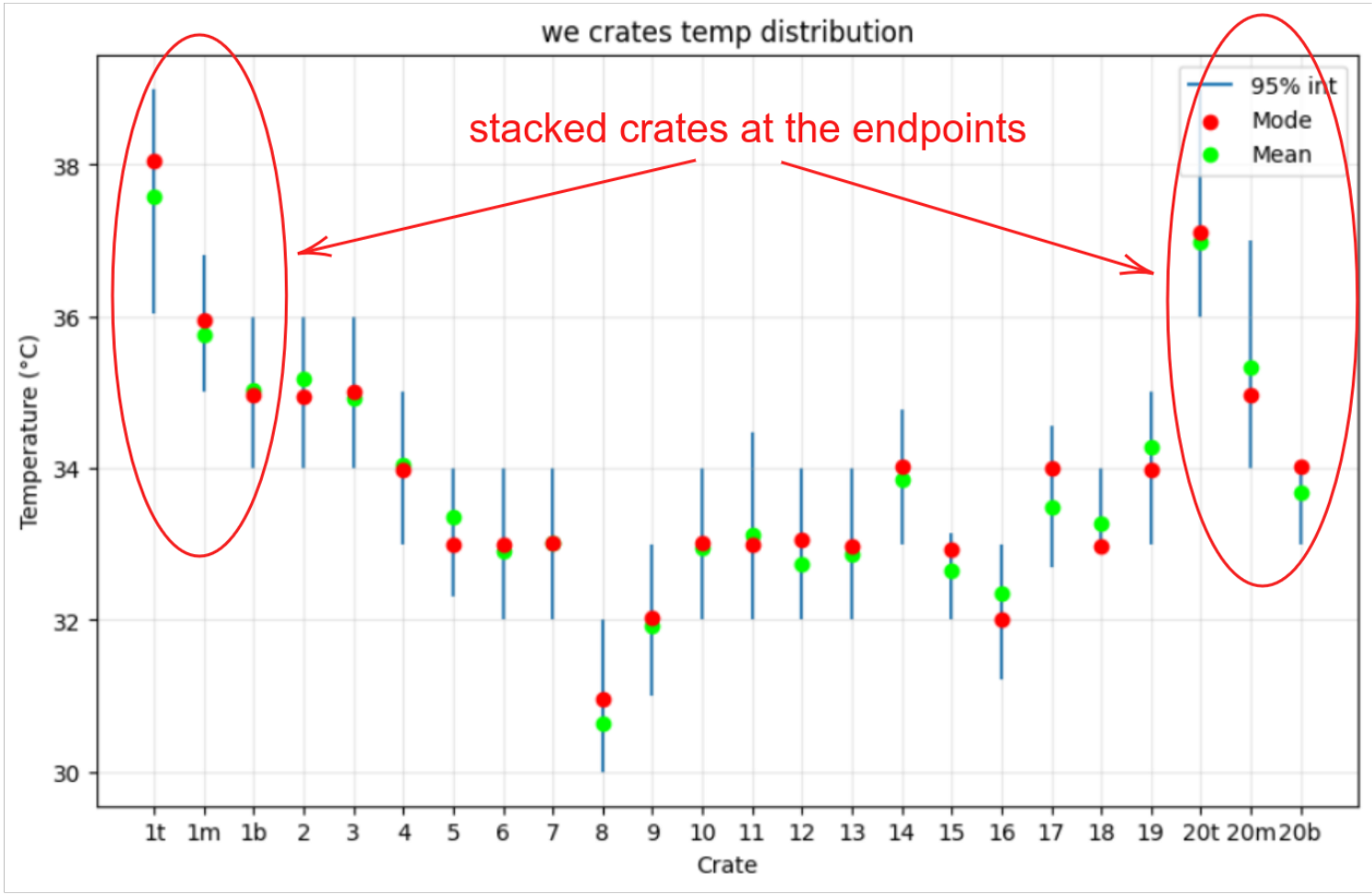


Crates top view



Row-of-crates temperature distribution - Run 10235

The alarms at the crate level are substantiated by the data obtained from Run 10235. A common U-shaped behaviour of the crate's means and modes is noticeable: as one approaches the top crates at the edges of each row (01T and 20T) the average temperature of the crate increases.



Grafana alert rules - 2

The **temperature alerts** are structured in order to go through the following steps **every 2 minutes**:

1. Fetch the current temperature value for each board and check if the difference with the temperature value of 10 minutes back in time of the same board is above 1.5 °C
2. Check if during the past 20 minutes the run number has not changed
3. Fetch the last temperature value for each board and check if it is above 38 °C

Then, **an alarm is fired if the statement** $(1 \wedge 2) \vee 3$ **returns True**. If the alarm fires then a Slack notification is sent to `test_grafana_alerts`.

Crate temperature time series - Run 10265

Insights for these conditions come from the analysis of Run 10265: in $\sim$ 10 minutes (timestamps 10:30 to 10:40) most boards are subject to a temperature rise of 2 °C.

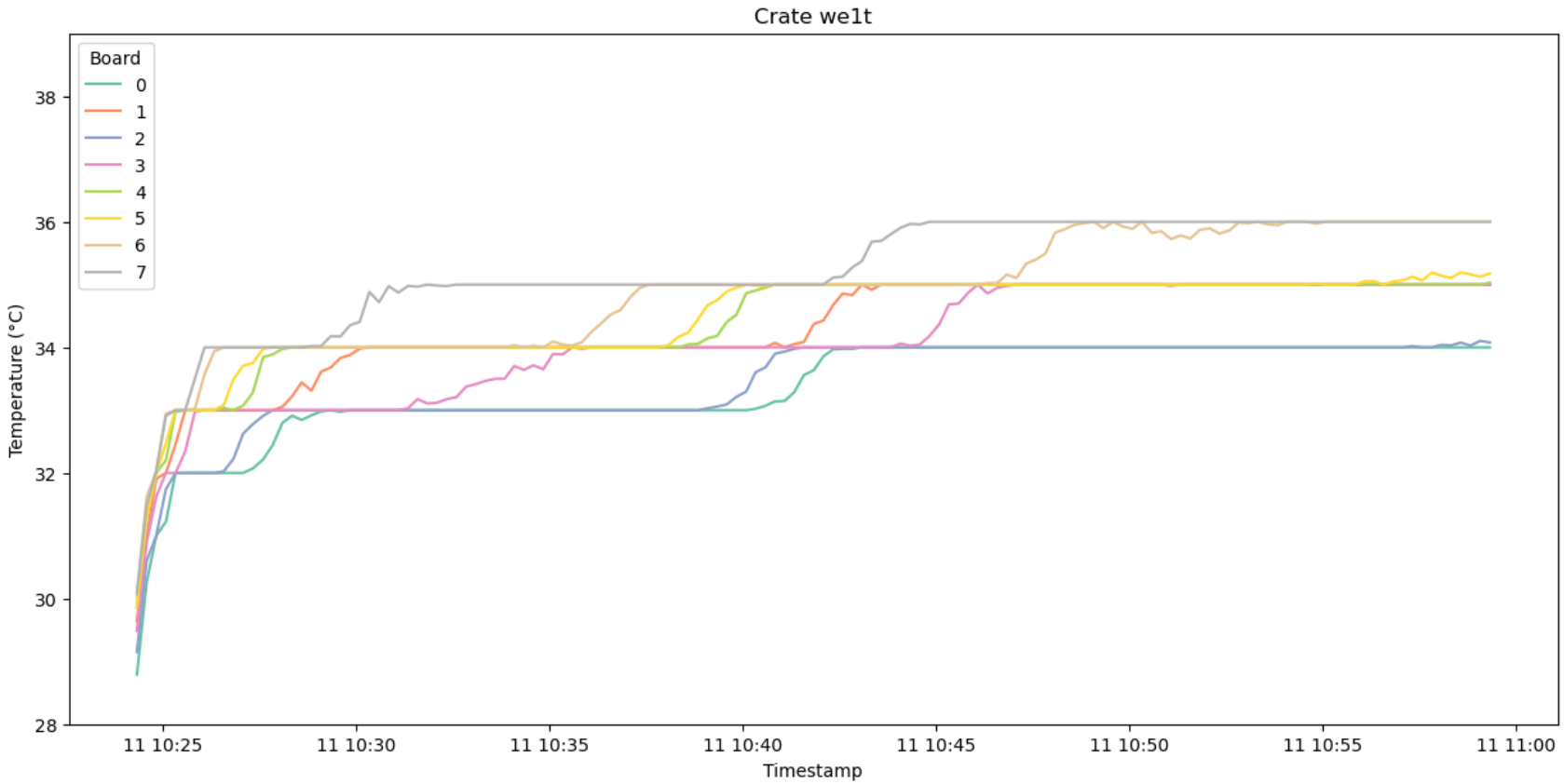
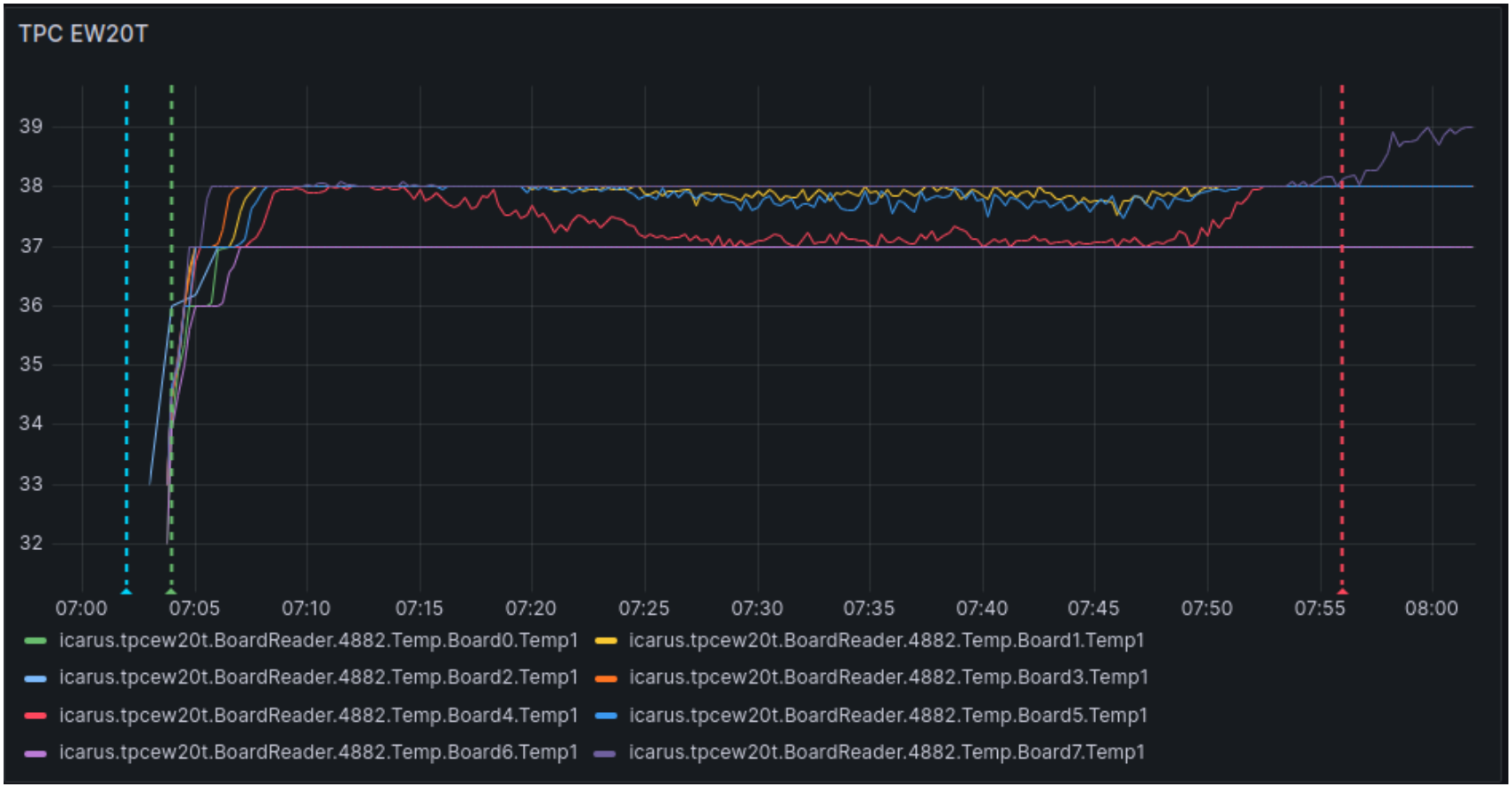


Figure: Run 10265 was started without mezzanine air conditioning, in order to "simulate" a rising temperature condition.

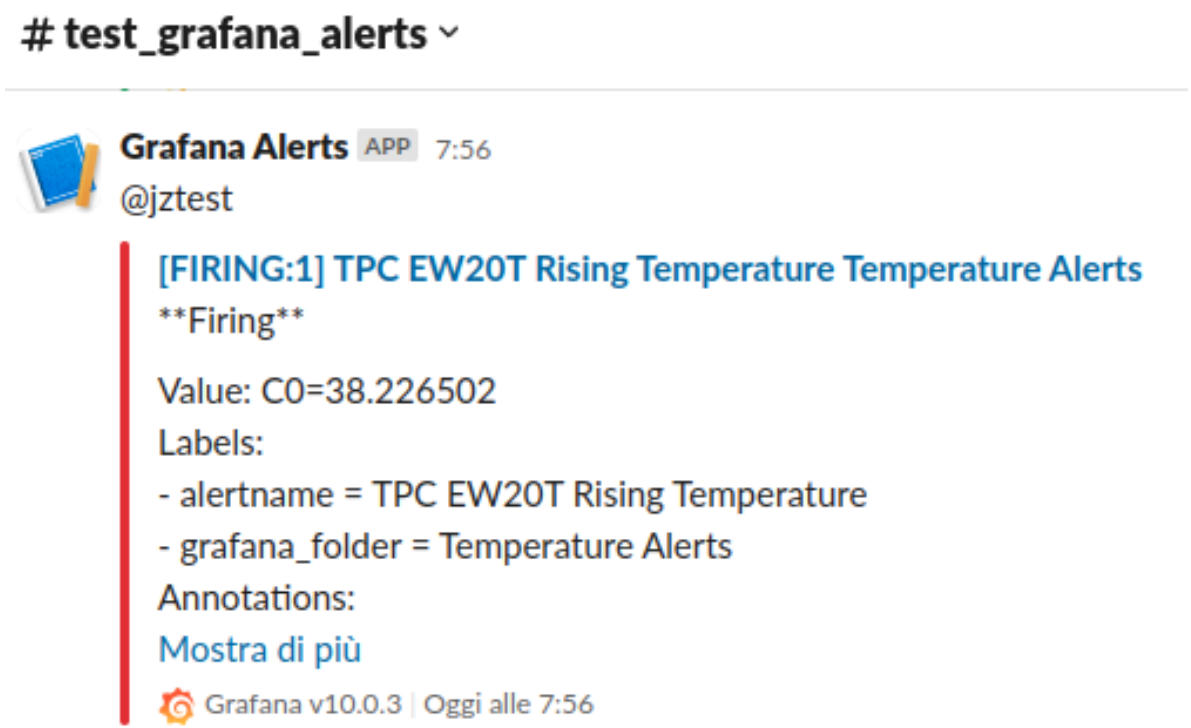
Grafana alert rules - 3

Here is, for example, an event occurred on Sep 24 in which Board7 broke the 38 °C threshold, causing the alert to fire and send a notification on `test_grafana_alerts`.



Grafana alert rules - 4

Here's the Slack notification sent on the `test_grafana_alerts` channel. Notice that the notification automatically reports the value that caused the rule to fire, even if it does not specify the precise metric (thus, which board) the value refers to.



Grafana alert rules - 5

- The **NoData alert** instead is simpler and checks every minute **if the last value** coming from any board within the crate **retrieves NoData**. A NoData value has to be **persistent for at least 5 minutes** in order to avoid firing an alarm if for any reason a occasional NoData value occurs
- If the alarm fires, then a Slack notification is sent to `test_nodata_alerts`

Grafana dashboard

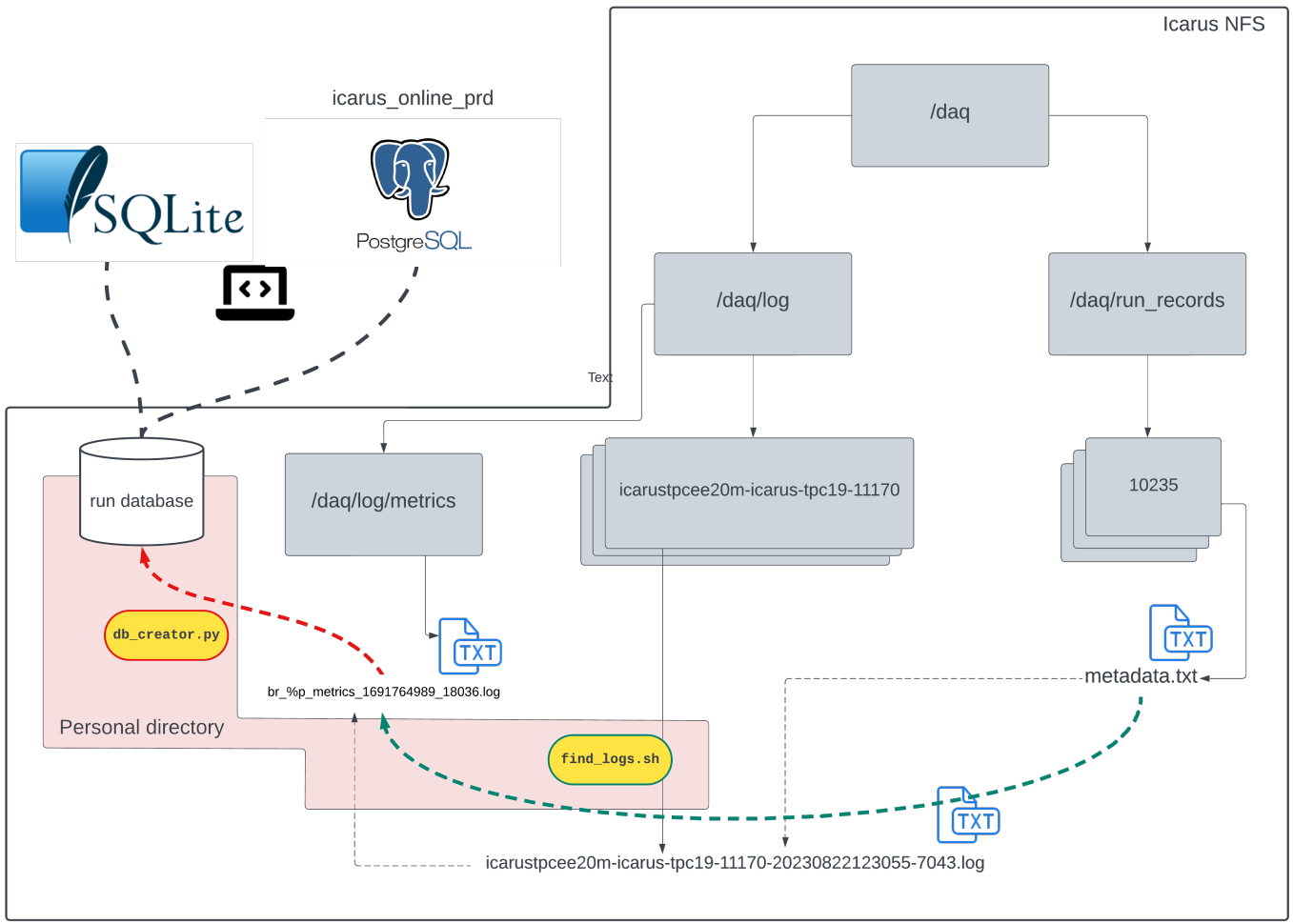
A Grafana dashboard with all 96 crates panels has been created by exploiting Grafana *provisioning* i.e. dashboards creation directly from configuration files. Provisioning features include also UI shortcuts that allow editing dashboards from `.json` files ($\rightarrow$ faster and more reliable) instead of UI dialogs.

Presentation overview

1. **Discussion on Grafana:** alerts, dashboards and panels
2. Solutions for **middle-long term temperature monitoring**

Solutions for temperature monitoring - 1

In order to perform the temperature characterization, **uncompressed and granular** board temperature **data** has been fetched directly **from the metric log files** generated by the `BoardReader` process



Solutions for temperature monitoring - 2

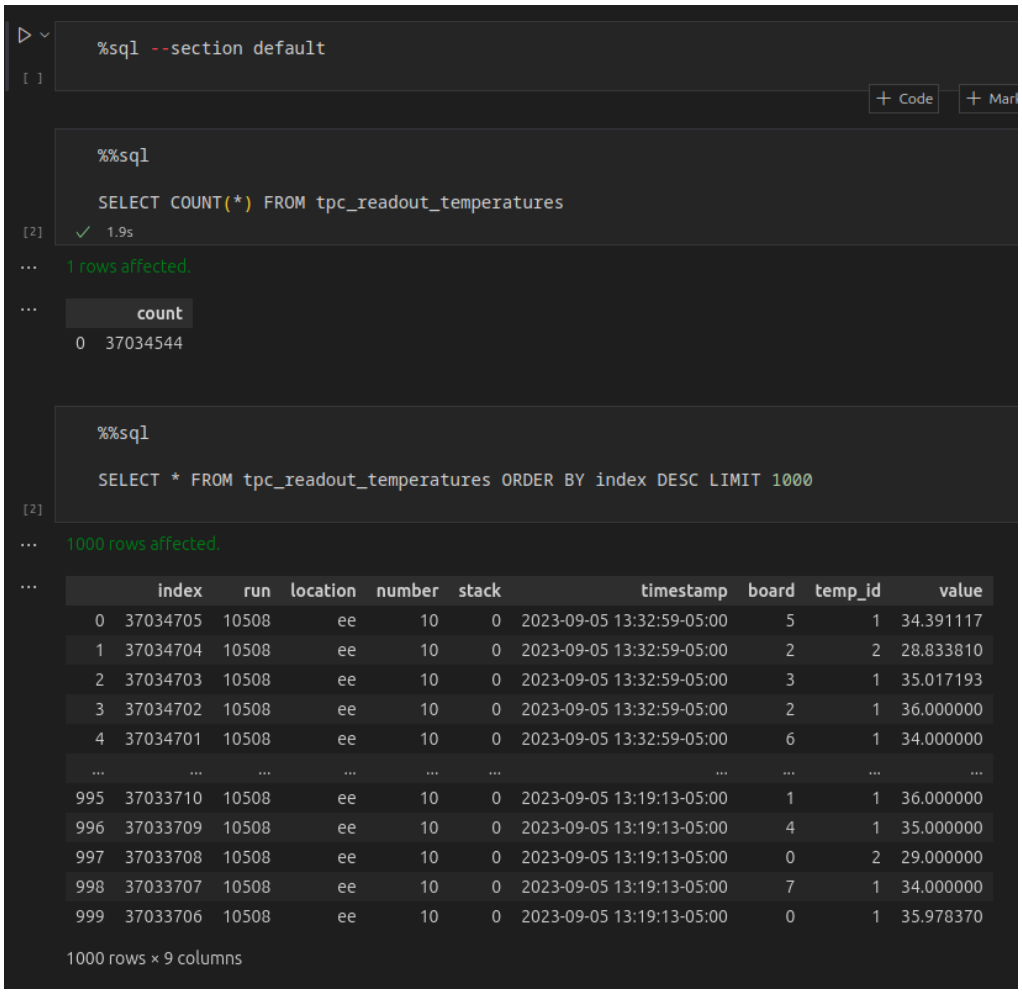
The workflow can be **summarized in the following points**:

- Bash and Python scripts provide fetching and parsing all the metric log files for a given run respectively
- According to the version of the scripts the data can be fed to a sqlite database file or inserted into an online PostgresDB (`icarus_online_prd` database) that is replicated to the `ifdb9` database to ensure offline acces
- The scripts are embedded in a larger Bash script that is intended to **run as a cronjob** inside `icarus-dcs01`
- **The cronjob provides basic error-handling** and a **logging system** to keep track of the status of the cronjob

```
191 find_logs: Inspecting boardreader log file /daq/log/icarustpcwe05-icarus-tpc11-11203/icarustpcwe05-icarus-tpc11-11203-20230915171622-11681.log
192       Run: 10572, Location: we, Number: 05, Stack: 0, Metric path: /daq/log/metrics/br_%p_metrics_1694816386_11681.log added to metric_paths_run_10572_v3.txt
193 find_logs: Inspecting boardreader log file /daq/log/icarustpcwe06-icarus-tpc11-11204/icarustpcwe06-icarus-tpc11-11204-20230915171622-11682.log
194       Run: 10572, Location: we, Number: 06, Stack: 0, Metric path: /daq/log/metrics/br_%p_metrics_1694816386_11682.log added to metric_paths_run_10572_v3.txt
195
196 -------------------- Fetching log files of Run 10572 Completed --------------------
197
198
199 -------------------- Loading records of Run 10572 into Database --------------------
200
201 ------------------------------------------------------------------------------------------------------------------------------------------------------
202 Parsing file #1 No. valid files before #0 ::: Run: 10572; Location: ww; Number: 11, Stack: 0, Path: /daq/log/metrics/br_%p_metrics_1694816386_25591.log
203 logfile_parser_script: Parsing time: 0.7263634204864502 s
204 logfile_parser_script: Data of metric log file /daq/log/metrics/br_%p_metrics_1694816386_25591.log loaded into database successfully ::: 15048 rows added
205 logfile_parser_script: Appending time: 4.628537893295288 s
206 ------------------------------------------------------------------------------------------------------------------------------------------------------
```

Accessing the database

Here are some considerations regarding accessing the data inside `icarus_online_prd`:



```
%sql --section default
```

+ Code + Mar

```
%%sql

SELECT COUNT(*) FROM tpc_readout_temperatures
```

[2] ✓ 1.9s

1 rows affected.

	count
0	37034544

```
%%sql

SELECT * FROM tpc_readout_temperatures ORDER BY index DESC LIMIT 1000
```

[2]

1000 rows affected.

	index	run	location	number	stack	timestamp	board	temp_id	value
0	37034705	10508	ee	10	0	2023-09-05 13:32:59-05:00	5	1	34.391117
1	37034704	10508	ee	10	0	2023-09-05 13:32:59-05:00	2	2	28.833810
2	37034703	10508	ee	10	0	2023-09-05 13:32:59-05:00	3	1	35.017193
3	37034702	10508	ee	10	0	2023-09-05 13:32:59-05:00	2	1	36.000000
4	37034701	10508	ee	10	0	2023-09-05 13:32:59-05:00	6	1	34.000000
...	...	...	...	...	...	...	...	...	...
995	37033710	10508	ee	10	0	2023-09-05 13:19:13-05:00	1	1	36.000000
996	37033709	10508	ee	10	0	2023-09-05 13:19:13-05:00	4	1	35.000000
997	37033708	10508	ee	10	0	2023-09-05 13:19:13-05:00	0	2	29.000000
998	37033707	10508	ee	10	0	2023-09-05 13:19:13-05:00	7	1	34.000000
999	37033706	10508	ee	10	0	2023-09-05 13:19:13-05:00	0	1	35.978370

1000 rows × 9 columns

- Jupyter Notebooks running directly on a users' PC can be used to perform further analysis that rely on the data fed to the PostgresDB
- Jupyter Notebooks can be used also to perform hot-fixes to the database whenever it may be necessary
- In this context, the JupySQL Python library results very useful thanks to its features, e.g. creating a connection file with several database connection credentials listed, or the possibility of using the `%sql` and `%%sql` magics for inspecting the databases

Next steps / improvements

- **Discuss** the temperature **alerts parameters**, taking into account
  - Sensitivity of the temperature from the location of the vents on top of the cryostat. Indeed the alerts shown in this presentation have been active since Sep 6 and have not fired until Sep 13, following a minor rearrangement of the vents
  - The boards maximum temperature allowance (40 °C)
  - During the winter, ambient temperatures may be lower
- Improve the "look" of the Slack notifications by adding tags or customized text with useful information, thus overriding the existing default schema
- Improve the reliability, the efficiency and the error handling of the cronjob. Moreover, discuss the data retention period.
- Study the correlation between the TPC board temperatures and the power supplies temperatures
- Perform a new characterization of the board temperatures with the presence of the beam and the lower temperatures during the winter

Thank you for your attention!

Giovanni Zago

PoD Master's Degree Student, Università di Padova

Email: giovanni.zago.2@studenti.unipd.it

References

- ICARUS collaboration "ICARUS at the Fermilab Short-Baseline Neutrino Program – Initial Operation", arXiv:2301.08634
- Pedro A. N. Machado, Ornella Palamara, David W. Schmitz "The Short-Baseline Neutrino Program at Fermilab", arXiv:1903.04608