Queue-based job monitoring

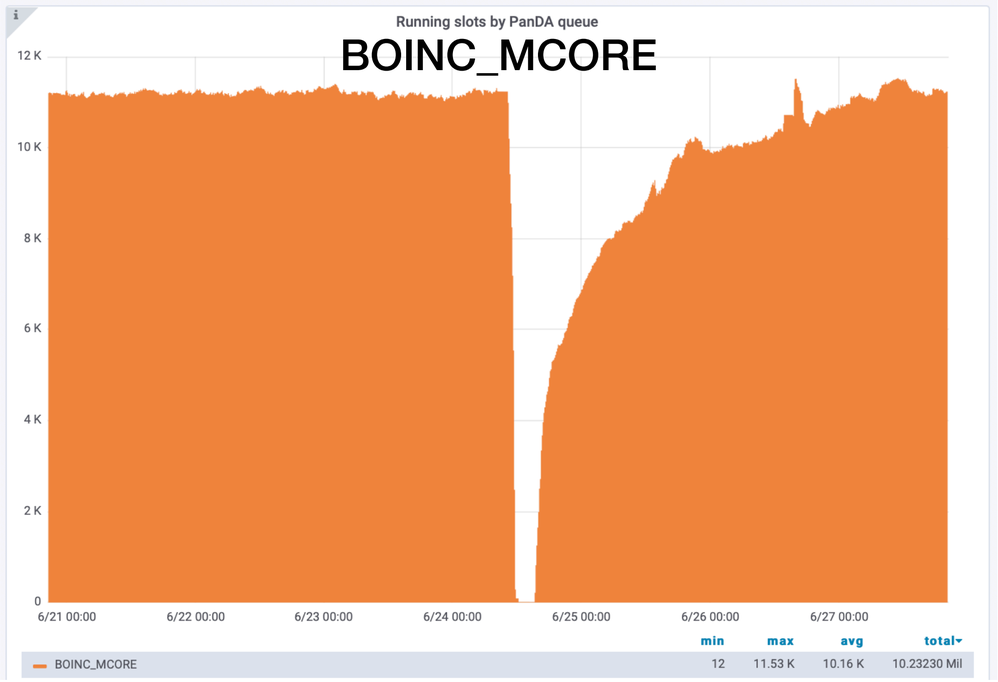
eric.schanet@cern.ch

Eric Schanet | LMU Munich | 30.09.2019

Introduction

- **Central problem:**
  - Why **specific variations in the numbers of running slots**?
  - Want to make sure that we are not wasting resources due to e.g. one FTS server not working as expected.
  - Need to make sure we detect problems as fast as possible.

Running slots over 7 days



- **Complex and multi-dimensional:**
  - Need to correlate things like e.g. pilot submissions, downtimes, failures, transfers, …, with e.g. activated and running jobs.

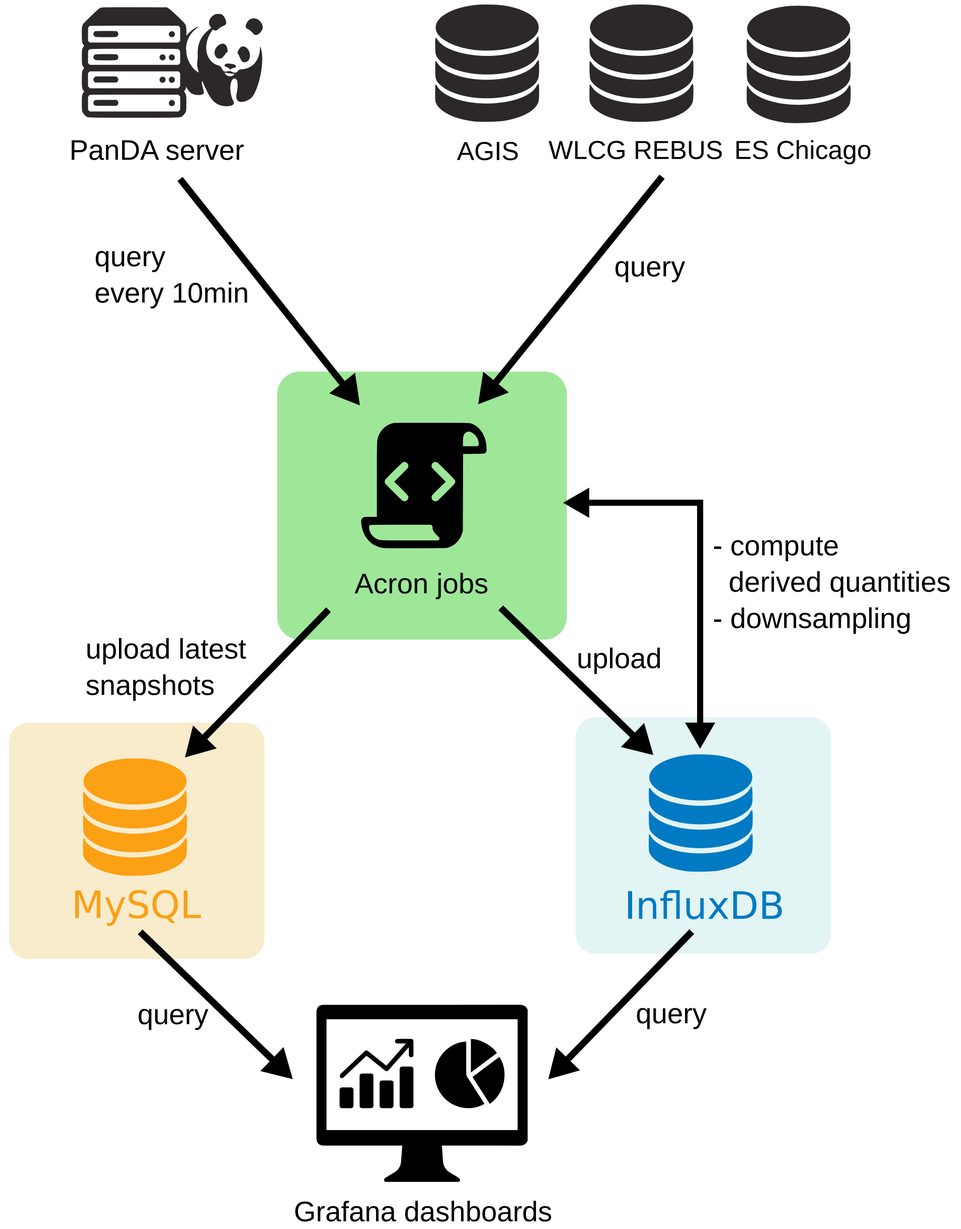
- **General idea:**
  - **Lightweight** and **low-latency**: allows to be near realtime and highly flexible without needing too much resources.
    ➡ Existing services are either too complex and heavy-duty or don't offer all the desired information.
  - **In principle all the information is already available** (somewhere), only need to combine and correlate it, but also display in smart and easy-to-digest way.

Technical setup

- **Main database: InfluxDB, 300GB size**
  - Filled with job information from PanDA, every 10min.
  - Only computing-site-level data, no job-level data (low latency!).
  - Information from other systems containing important metadata.
  - Downsampling (handled offline):
    - 10min granularity for 7 days,
    - 1h granularity for 2 months,
    - 1d granularity for 1 year.
  - Derived quantities mostly computed offline (e.g. moving averages).
    ➡ Prevents need for expensive on-the-fly computation.

- **Second database: MySQL, 300GB size**
  - Used for non time-series plots or tables.
  - Contains latest snapshots of InfluxDB data.
  - Keeps some more load off InfluxDB.

- **Cron jobs:**
  - All the downloading (writing) from (into) DBs is handled by simple scripts as cron-jobs.



Data structure in InfluxDB

- **From PanDA (distributed production and analysis system used by ATLAS):**
  - Number of jobs per PanDA queue, no job-level information! (ATLAS computing sites usually consist of multiple PanDA queues)
  - For each PanDA queue: One data point per job state and per resource type (SCORE, MCORE, etc.).

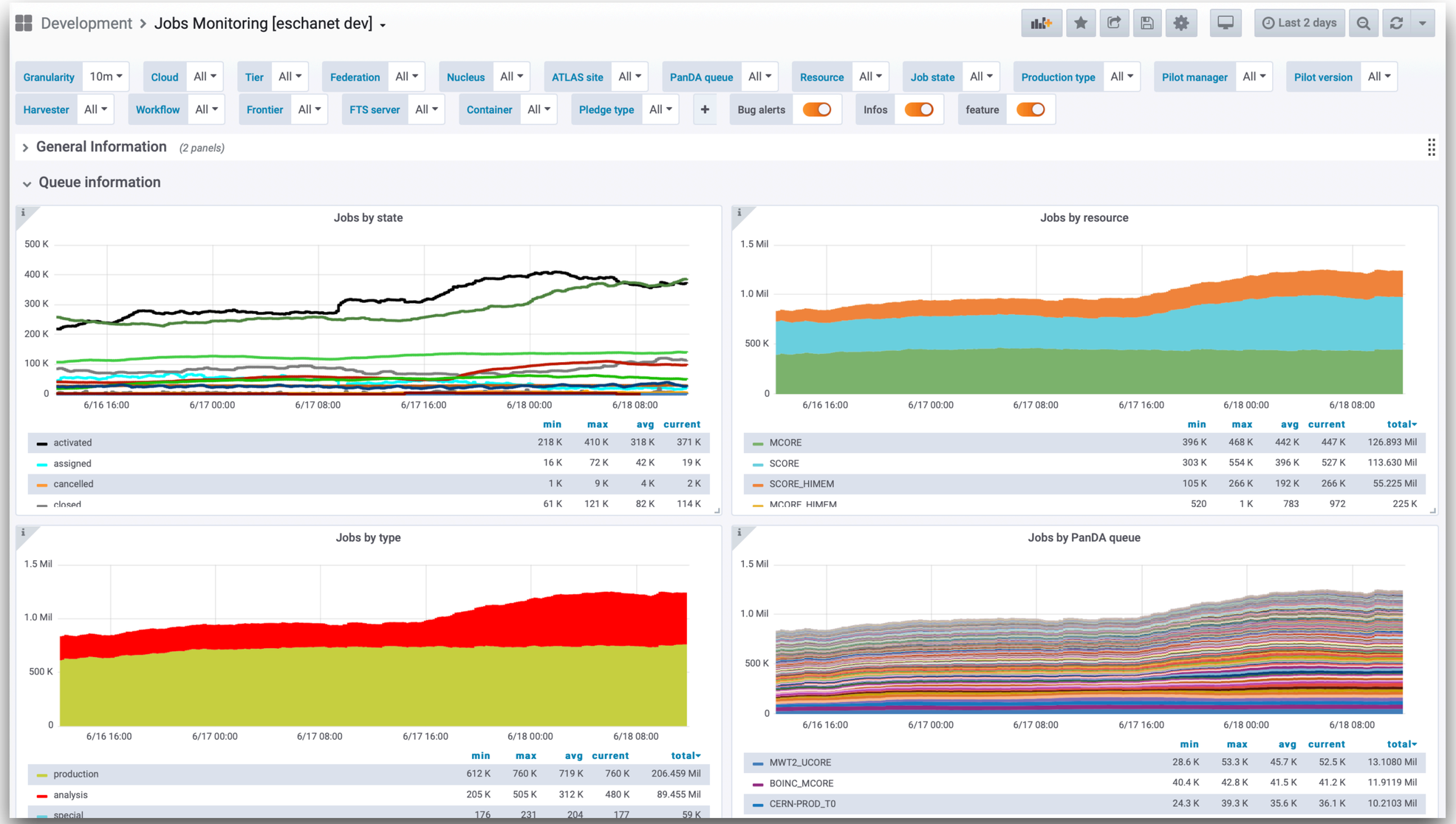
- **From ATLAS Grid Information System (AGIS):**
  - For each PanDA queue: Cloud, Tier, Nucleus, ATLAS site, production type, pilot manager, pilot type, harvester instance, harvester workflow, frontier, FTS server, container mode.

- **From WLCG Resource, Balance & Usage (REBUS):**
  - For each PanDA queue: Federation, pledge and pledge type of federation.

- **From other ATLAS-internal sources**
  - Benchmark job results (measured HS06) per PanDA queue.

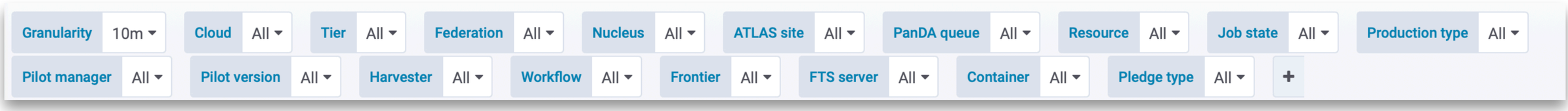
Monitoring: Overview

- Queue-based low-latency job monitoring dashboard built with InfluxDB and Grafana (hosted by CERN).

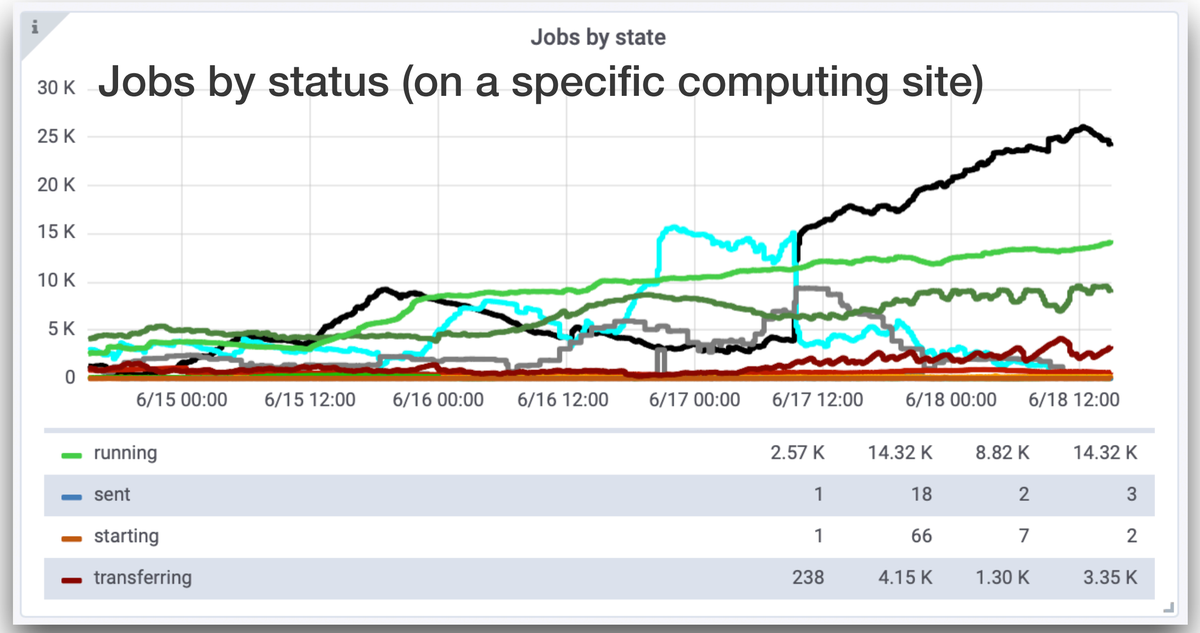


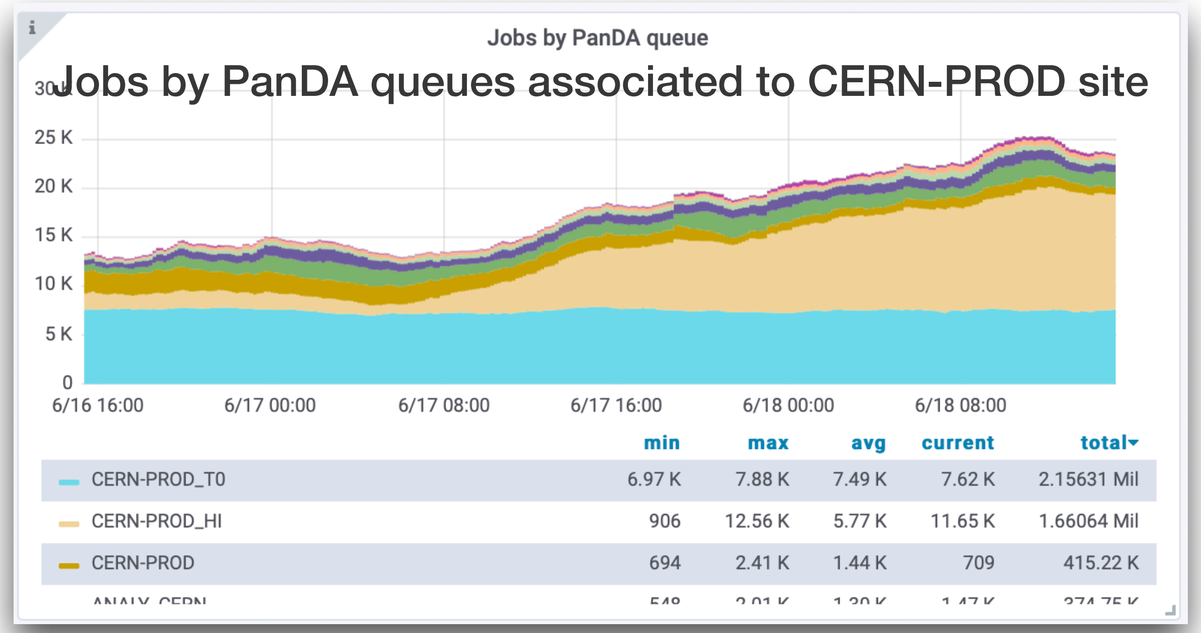
Monitoring: Examples

- **Low latency, high granularity plots showing numbers of jobs**
  - Highly interactive, users can e.g. freely choose time-range (Grafana!)
  - Large number of parameters users can choose from:



- **Only few of the many plots that can are available :**



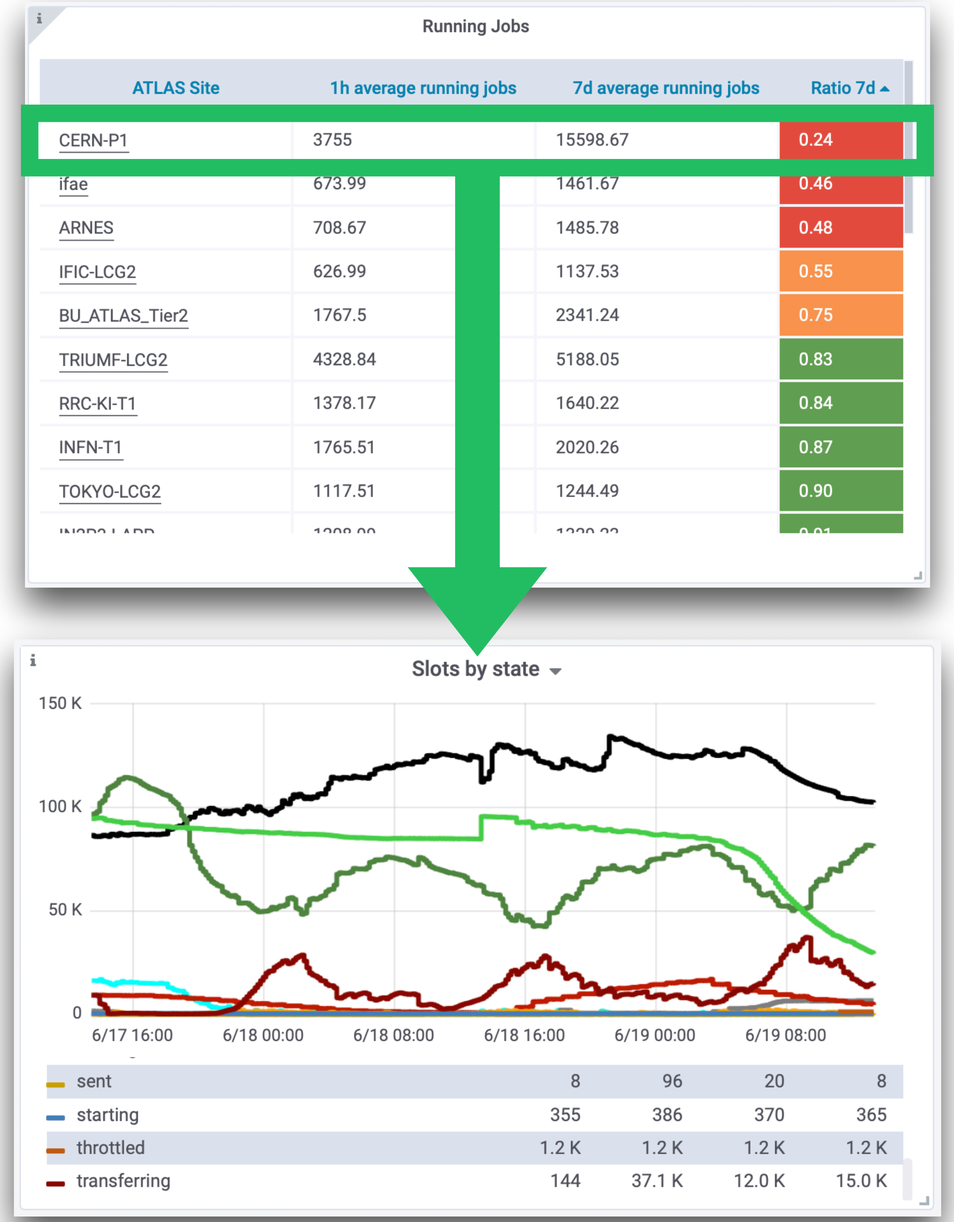


Detecting sudden changes

- **Low-latency perfect for detecting sudden changes, empty queues, etc.**
  - Suspicious sites dashboard
  - Very simple setup, aims to detect problematic sites by flagging sudden changes compared to a user-defined period of time.

- **Example on the right:**
  - Top table: showing ratios between running jobs of 1h and 7d.
  - CERN-P1 shows relatively low ratio (only 25% of running jobs currently compared to last 7d).
  - Click on CERN-P1: job monitoring page for site is opened.
  - ➡ Grafana allows to integrate many different dashboards and views with each other.

- User can choose from large set of parameters to confine results
  - Selections are possible, e.g. looking only at Tier0+Tier1 sites.



Summary and outlook

- **Queue-based monitoring using Grafana and an InfluxDB hosted at CERN:** job monitoring dashboard
  - Low-latency and lightweight service offering interactive monitoring dashboards/plots .
  - Based on queue-level data from PanDA.
  - Integrates information from AGIS, REBUS and other ATLAS-internal DBs.

- **Based on monitoring plots:** automatic detection of problematic sites
  - Very simple example: suspicious sites dashboard
  - Using simple metrics like e.g. ratios of average numbers of jobs in order to detect sudden changes.

- **Can this be used for other experiments?**
  - Setup is of course quite specific to ATLAS and would need changes to fit to other experiments …
  - PanDA can in principle just be exchanged with any other production system that has some sort of http interface.
  - Doesn't need too much resources but still is able to provide low-latency monitoring combining different sources of information.