Performance monitoring of opportunistic resources at ATLAS-BFG

Michael Böhler, Anton J. Gamel, Stefan Kroboth*, Benjamin Rottler, Markus Schumacher

22.09.2020

Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

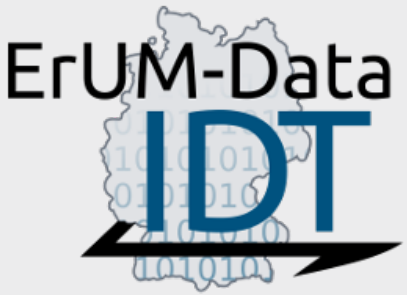


Table of Contents

- Setup at Uni Freiburg using COBaID/TARDIS
- Our contributions to COBaID/TARDIS
- Monitoring of opportunistic resources and related infrastructure
- Bigger picture / future plans

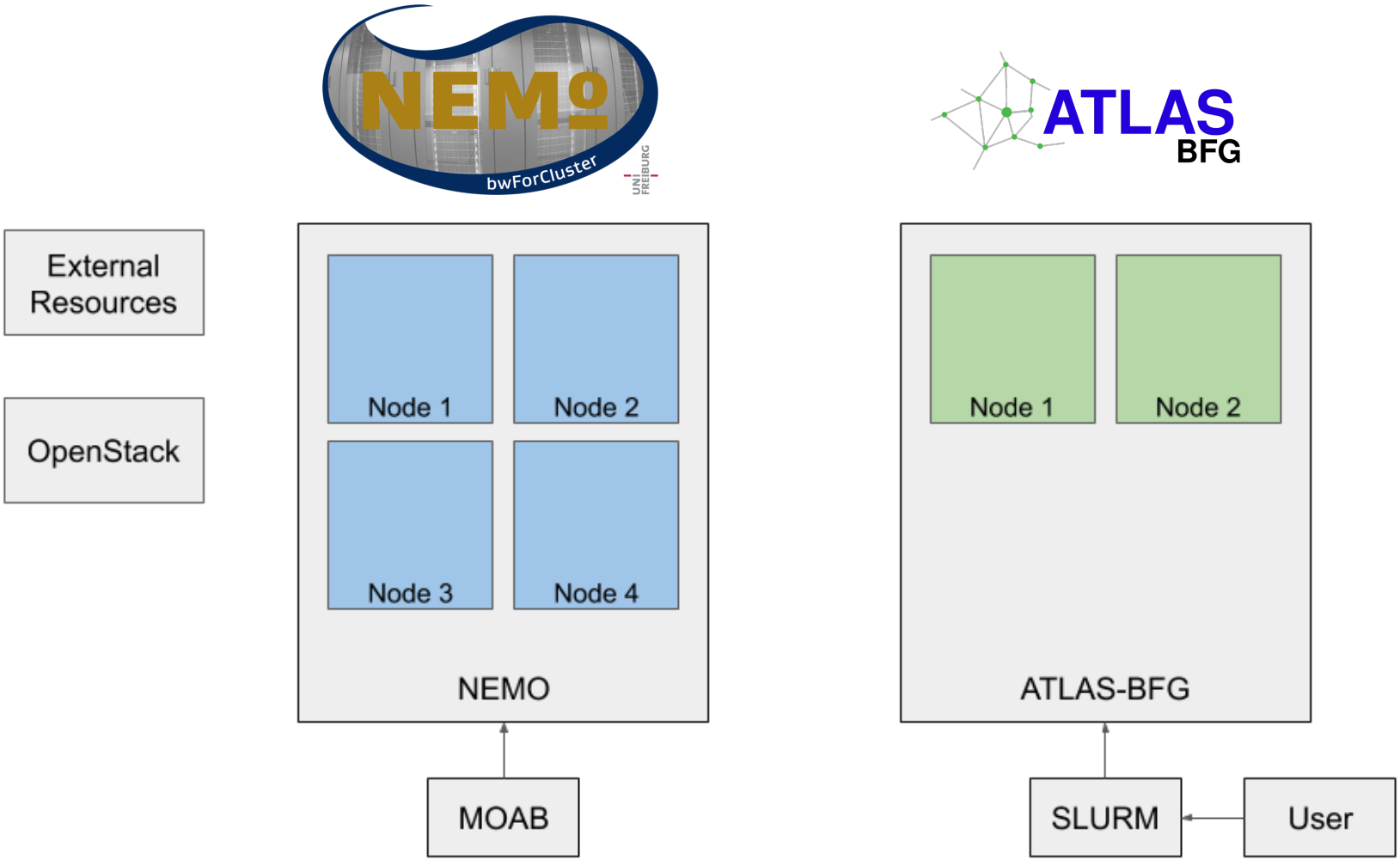
Setup / Infrastructure

Two clusters available:

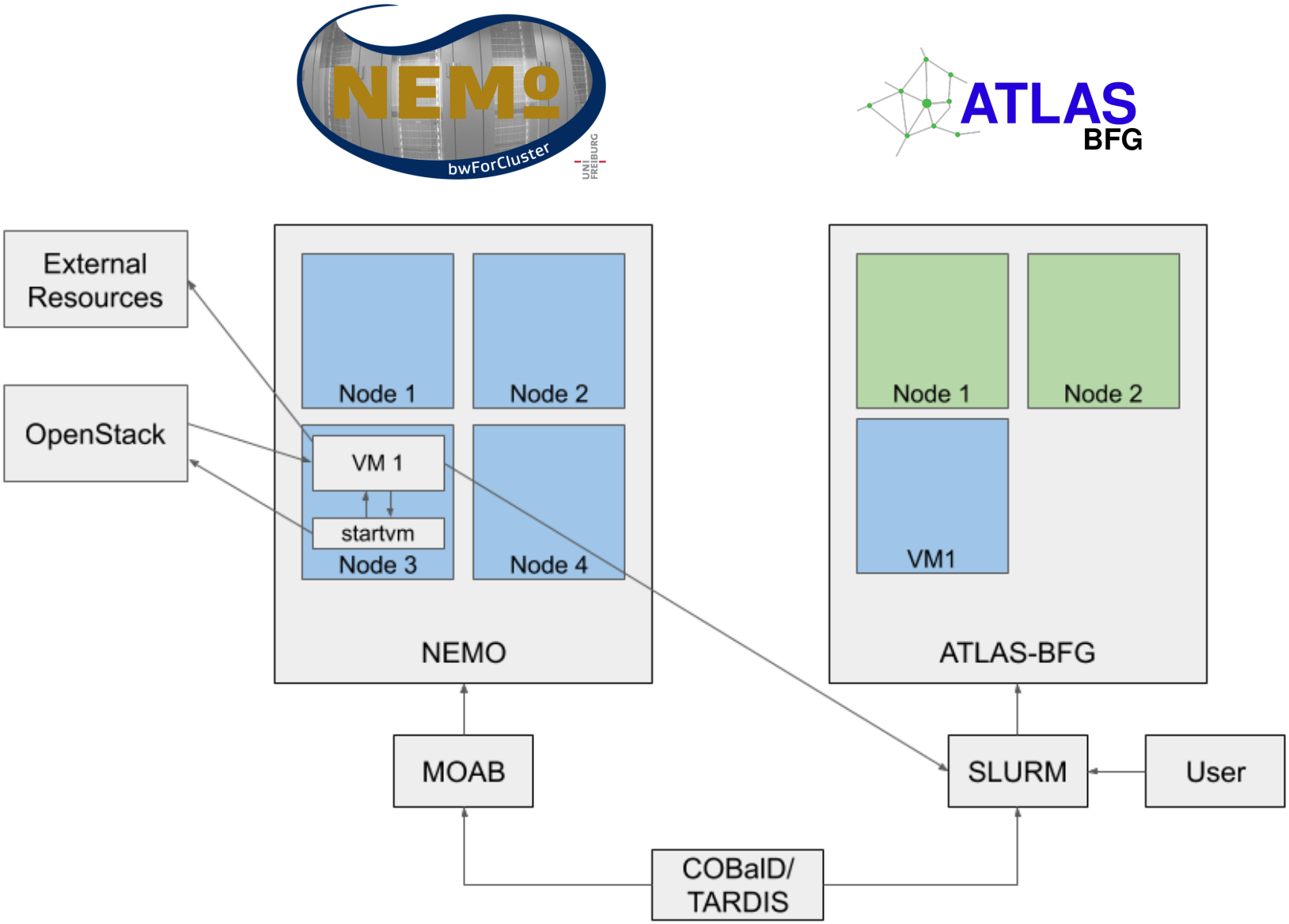
- ATLAS-BFG
  - $\approx$ 3600 cores
  - ATLAS production/analysis jobs
  - Local user jobs
  - Scheduler: SLURM
- NEMO
  - $\approx$ 18000 cores
  - Local cluster by Freiburg University
  - Different software setup than ATLAS-BFG
  - Scheduler: MOAB

Utilize COBaID/TARDIS to opportunistically integrate resources from NEMO into ATLAS-BFG based on demand and availability

Setup / Infrastructure



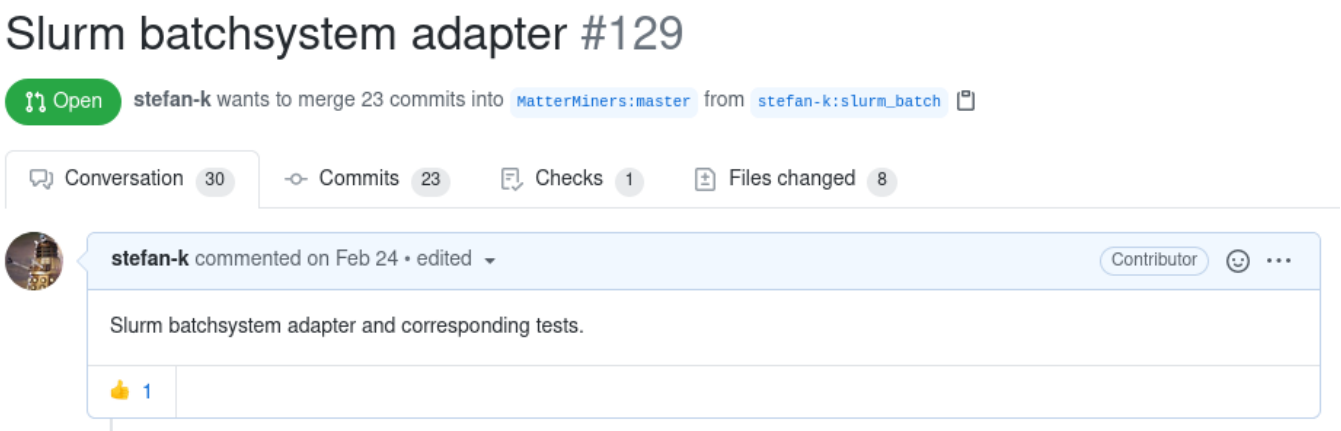
Setup / Infrastructure



COBaID/TARDIS

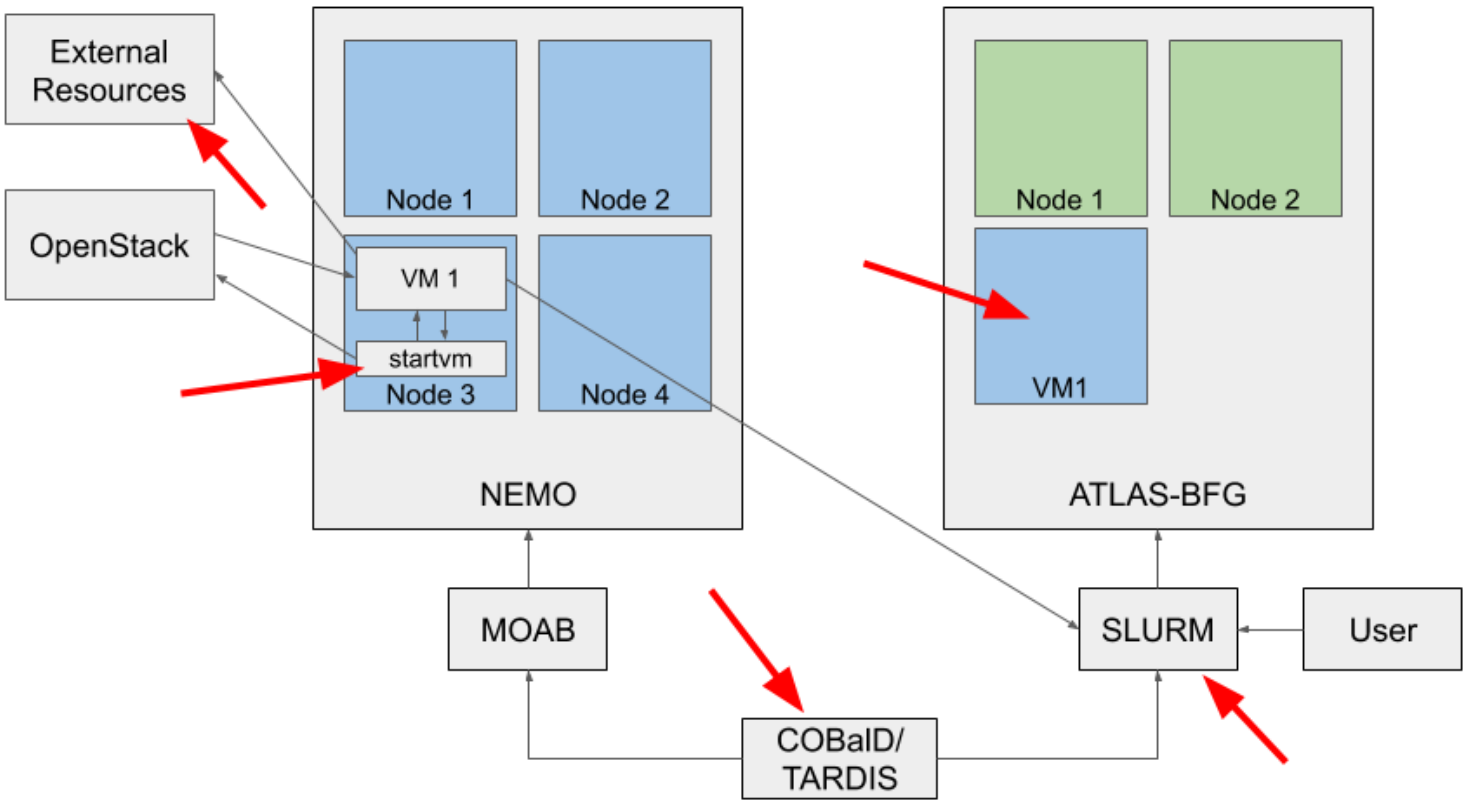
SLURM batch system adapter

- COBaID/TARDIS requires knowledge about the workload on the *drones* to make decisions about whether to start or stop drones which can be obtained from SLURM
- Therefore a SLURM batch system adapter which regularly queries SLURM was added to COBaID/TARDIS



Monitoring

- Complex infrastructure with many components and interactions
- Varying temporal dependencies
- Difficult debugging in case of issues
- Difficult to assess the impact on performance when changes are made



Monitoring

Software stack

Metrics storage

Prometheus

Structured data

elasticsearch

Alerting

ZABBIX

Visualization

Grafana

Utilities

Metrics exporters, Vector

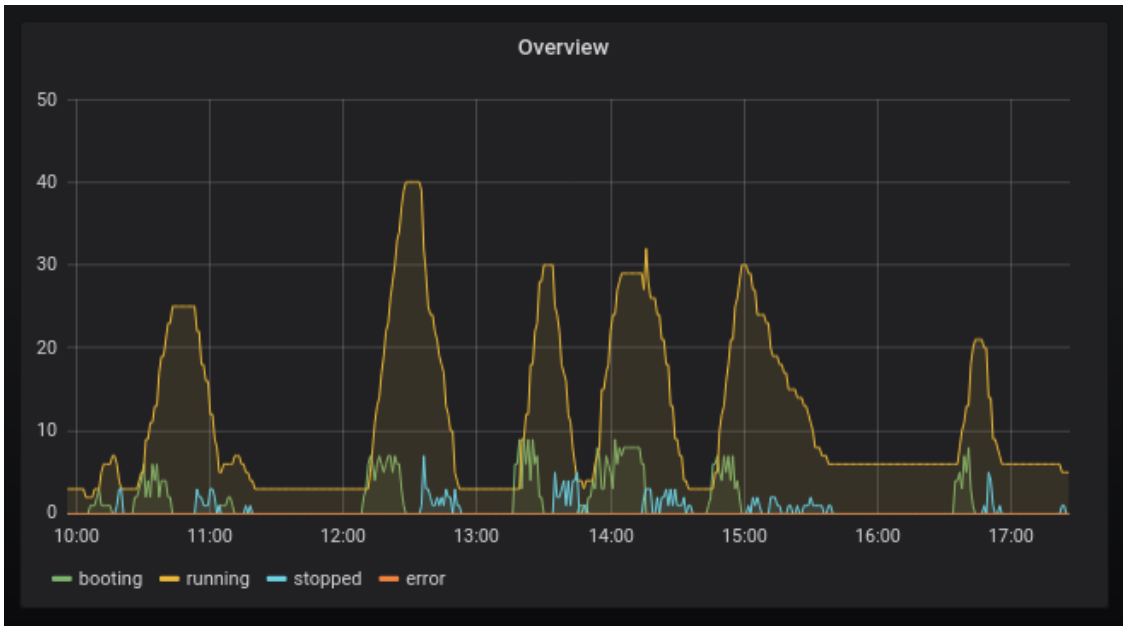
Elasticsearch monitoring plugin

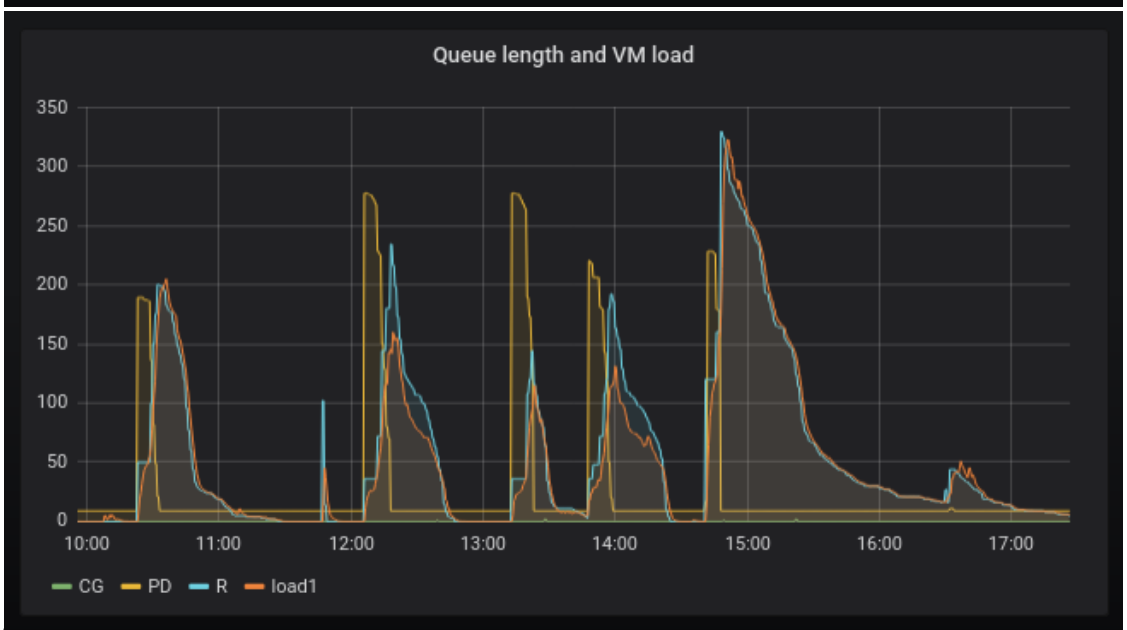
- Reports state changes of the drones to an Elasticsearch instance.
- This allows to continuously or retrospectively derive metrics

Prometheus monitoring plugin

- Keeps track of the current number of drones in each state
- Serves this information to the Prometheus scraper

Examples





- Top: Number of booting/running/stopped drones according to COBalD/TARDIS
- Bottom: Pending/running jobs according to SLURM
- Exposes temporal relationship between the individual components

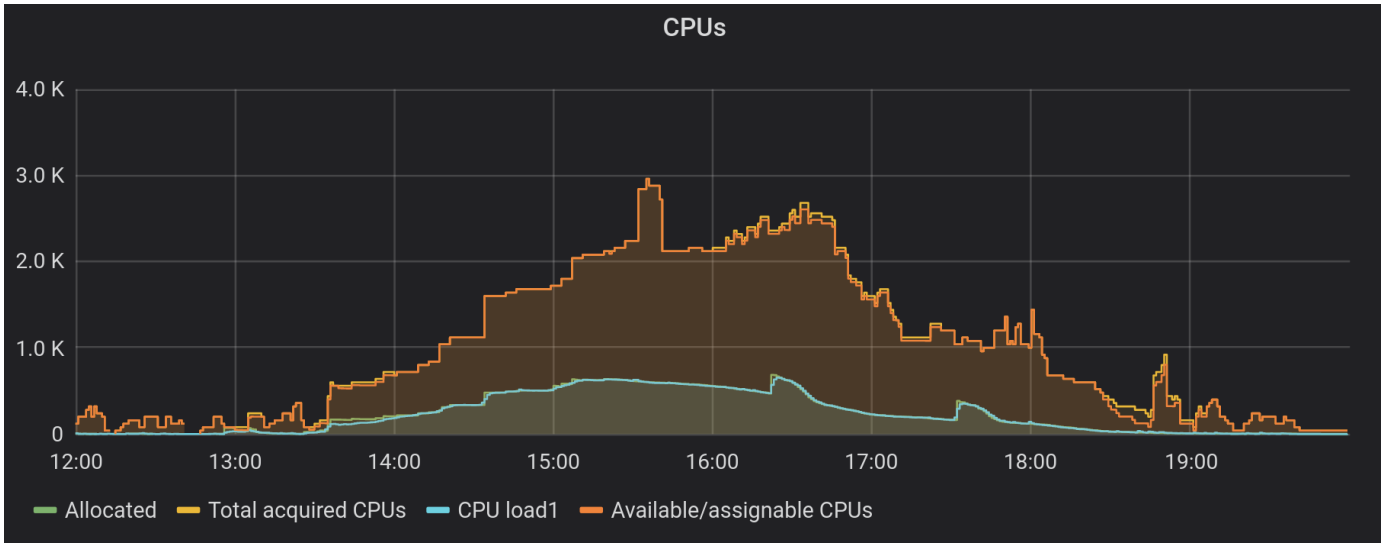
Examples

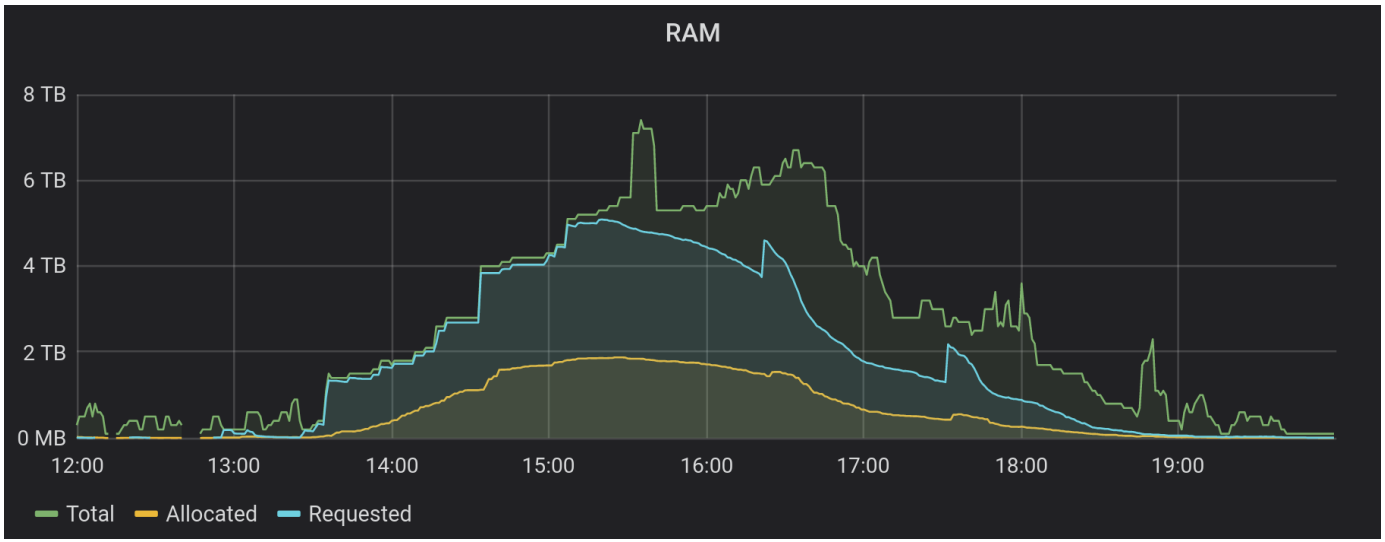
COBalD/TARDIS drone info

drone_uuid	state	machine_type	created	updated
nemo-8773904	IntegrateState	tardis_c40m100	2020-09-15T14:09:38.870313	2020-09-15T14:21:38.962304
nemo-8773901	IntegrateState	tardis_c40m100	2020-09-15T14:07:34.736392	2020-09-15T14:21:34.812700
nemo-8773894	AvailableState	tardis_c40m100	2020-09-15T14:05:29.397396	2020-09-15T14:21:29.586682
nemo-8773958	AvailableState	tardis_c40m100	2020-09-15T14:14:08.567953	2020-09-15T14:21:08.709344
nemo-8773993	AvailableState	tardis_c40m100	2020-09-15T14:15:55.361620	2020-09-15T14:20:55.479032
nemo-8773913	AvailableState	tardis_c40m100	2020-09-15T14:11:49.038027	2020-09-15T14:20:52.793295

Monitoring of every state change of the drones in real-time

Examples





- Top: acquired and available CPUs
- Bottom: acquired and available RAM
- Exposes inefficient use of the resources by the users
- Data can be used to inform such users automatically

Future developments

- Combine the collected data and derive meaningful metrics
  - Time from requesting a resource to resource being available
  - Measure for idle resources
- Monitor metrics over various time scales to detect drifts, sudden changes and anomalies
  - Aids in debugging and tuning
  - Increase overall performance and more efficient use of resources
  - Assess impact of job mix on performance
- Automatically give users feedback on their usage of the cluster
- Combine with alerting and event handling to automatically react to certain events