LIGO Monitoring and the Grafana Dashboard

Michael Thomas

California Institute of Technology
LIGO Livingston Observatory

Problems

Wanted a web-based tool for admins to monitor the health and status of a single condor cluster

- Need to be able to see different views of the cluster status
  - aggregate, per-schedd, per-user, per-searchtag, GPUs
- Need to see how these evolve over time
- Need to be able to quickly see which schedd's a user is submitting from
  - "My jobs won't run. Help!" "Which host are you submitting from?" "LLO" Doh!
- Don't want to remember all of the different CLI classad attributes and syntax
- condor_gangliad was not working out as we had hoped, including ganglia scaling issues

Problem solved!

Fifemon: https://research.cs.wisc.edu/htcondor/HTCondorWeek2016/presentations/ThuRetzke_Fifemon.pdf

Fifemon is a python service that uses the condor python bindings to gather htcondor metrics and dump the data into a time series database, visualized using grafana.

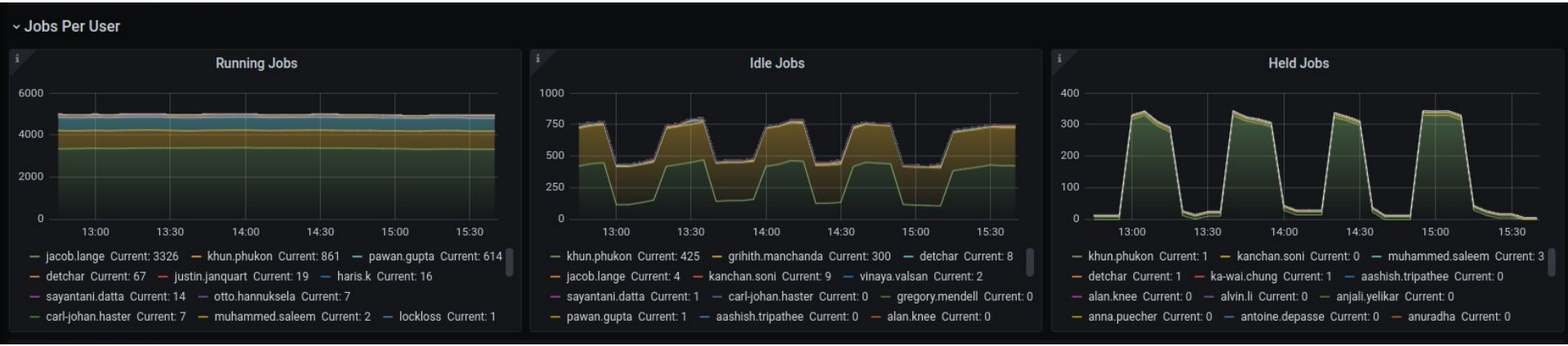
Some work needed

Probes needed some work to be usable in LIGO:

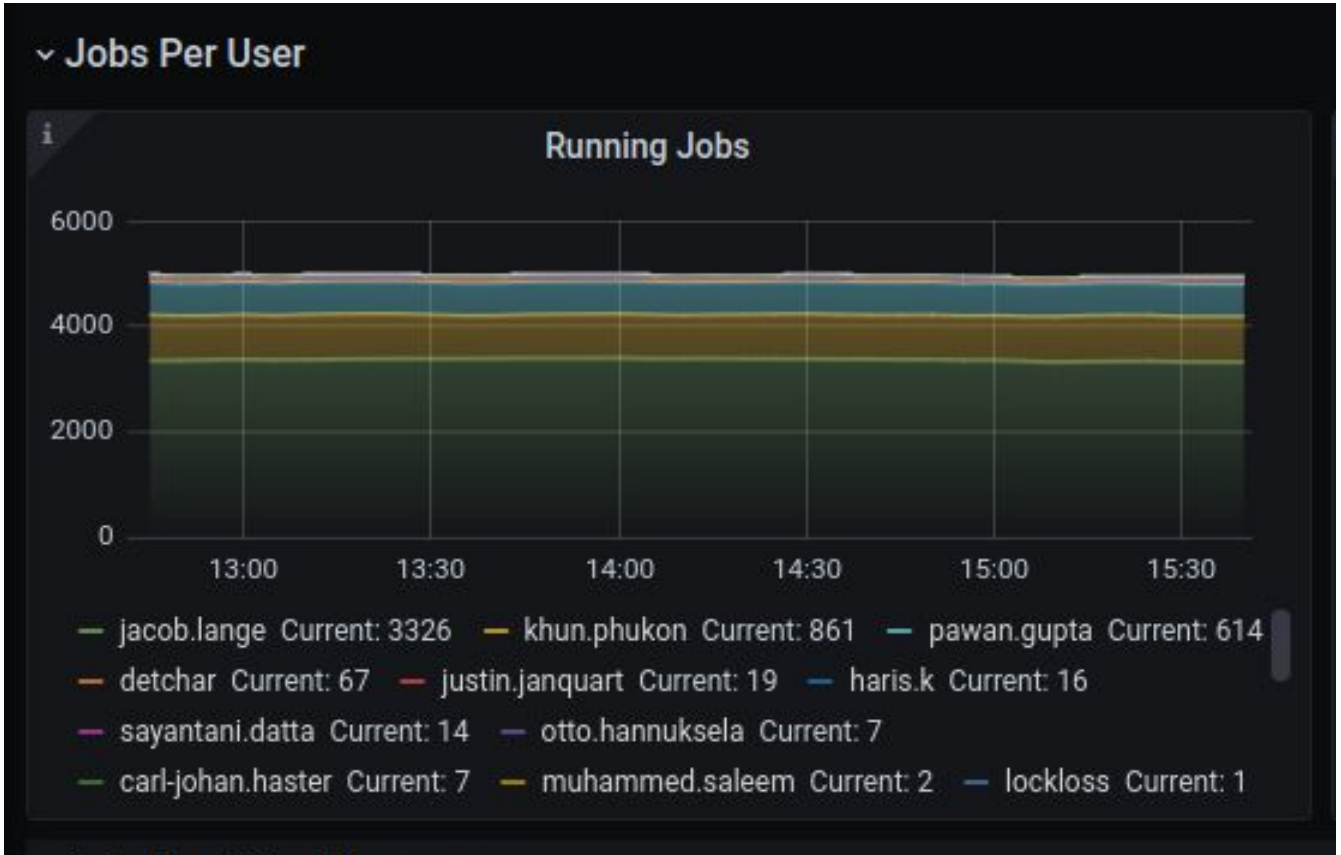
- Change AccountingTags to LigoSearchTag, removing 'experiment name' element in the hierarchy
- Allow a '.' in usernames
- Added GPU accounting
- Dashboard not quite functional with these changes
- Packaged as RPM for easier deployment
- All changes made public: https://git.ligo.org/michael.thomas/fifemon/-/compare/master...ligo

Cluster Overview

Jobs per user

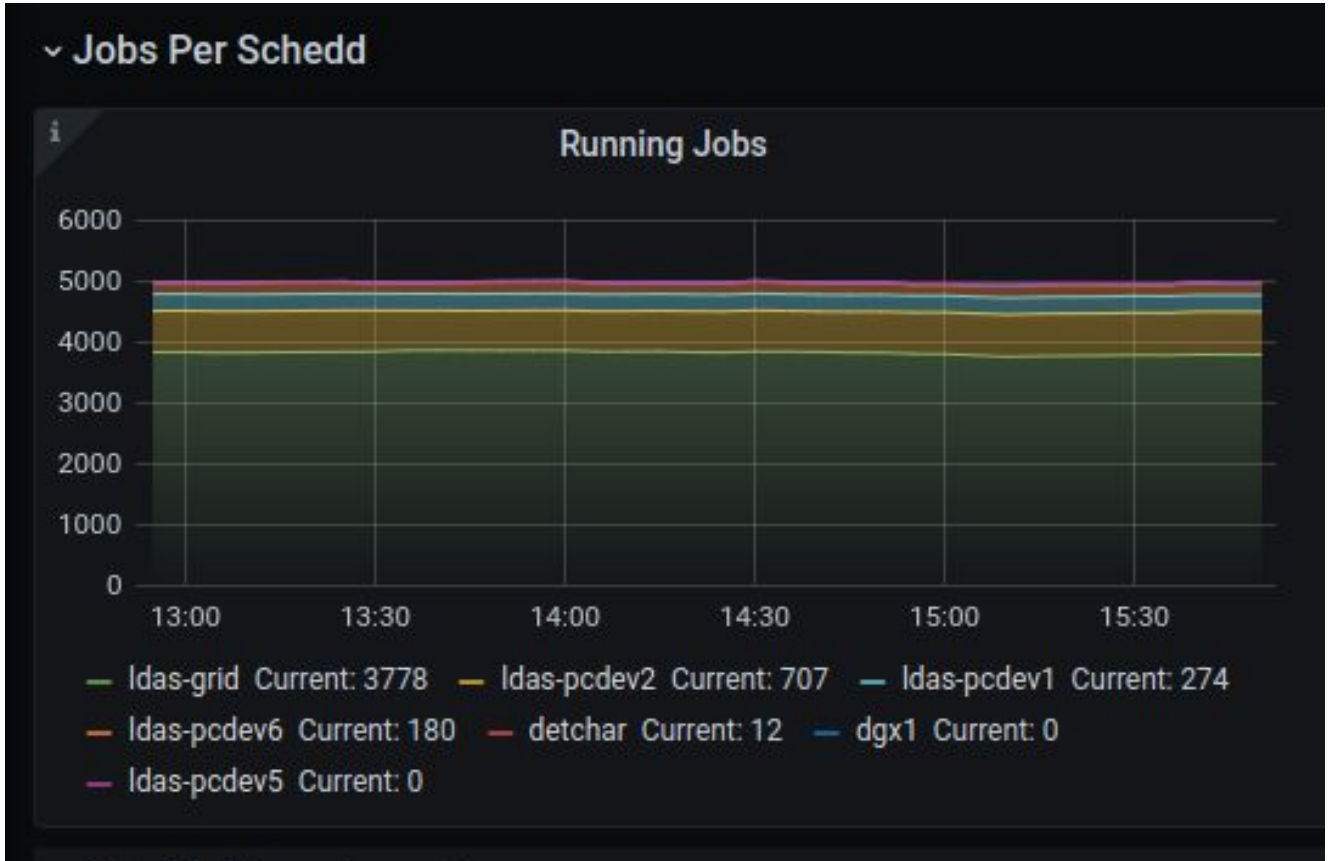


Jobs per user



Jobs per schedd

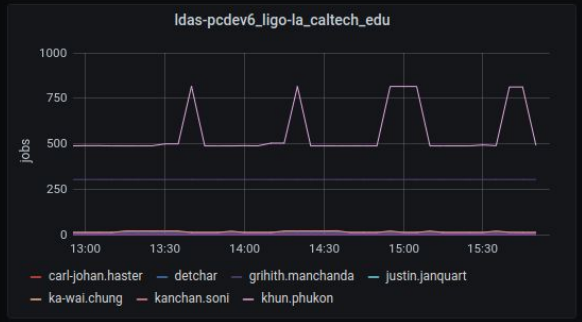
Jobs per schedd

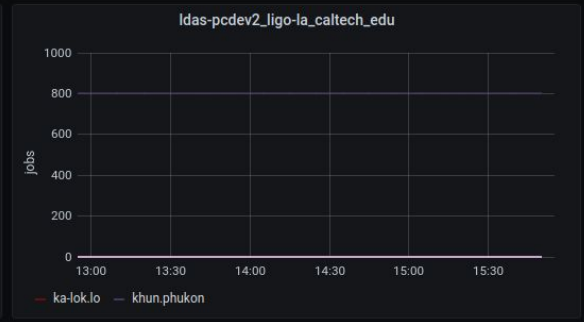


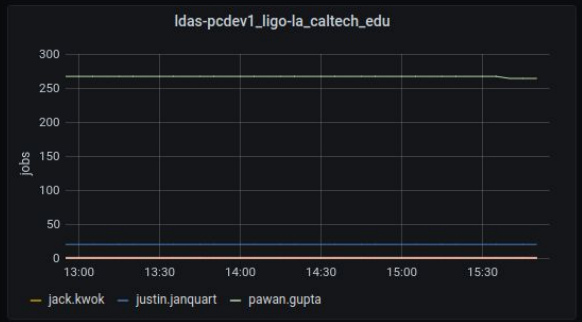
Condor

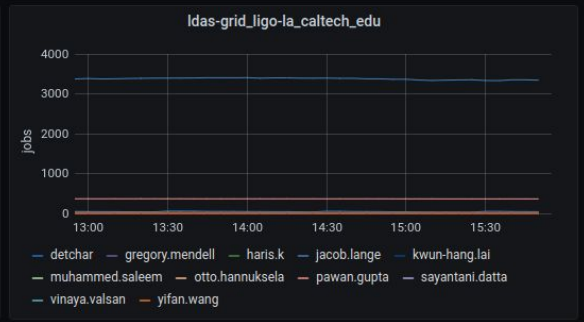
Last 3 hours | 1m

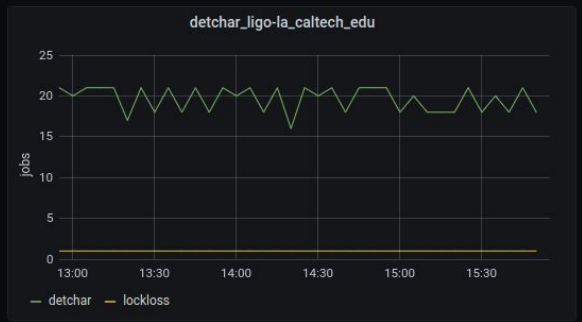
Users per Schedd



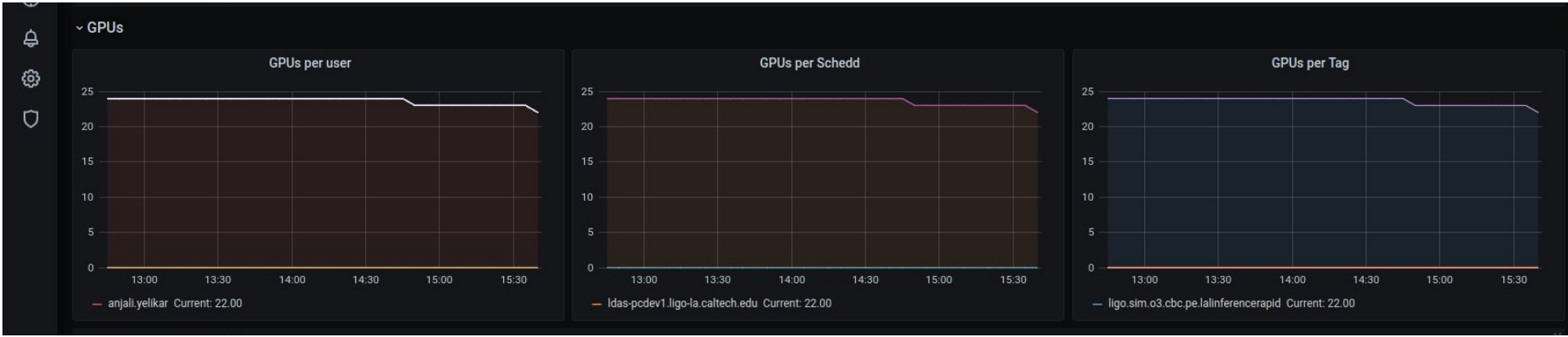








GPU monitoring



Other plots...

- Transfer queue wait time, number of upload/downloads (active and waiting)
- Job efficiency per user, tag
  - RemoteUserCPU / ((ServerTime-JobCurrentStartDate)*RequestCPUs)
- Total CPU, memory utilization
- Excess Requested (wasted) Memory (aggregate and per-user)
  - RequestMemory - ResidentSetSize_RAW
- Aggregate dynamic slots, CPU, Load, Memory, Disk Usage

How has it been: The good

- Immediately found issues with jobs on hold across multiple schedds
- Found schedds with suspiciously round number of jobs running. Needed to increase MAX_JOBS_RUNNING
- Found issues with memory requirements growing on jobs
- Found idle jobs when CPU, memory wasn't fully utilized
- Easy to get an overview of the batch system health after maintenance

How has it been: the bad

- Graphite overloaded
  - data gets dropped when too many metrics are collected in a single iteration (~10000)
  - data gets dropped when too many new metrics are collected in a single iteration (~500)
- Influx data storage never worked

Work to be done

- Deploy at LHO
- Add startd metrics
  - Help identify startds with free CPU, memory "Why do I have idle jobs when memory, cpu aren't saturated?"
- condor_metricsd?
- Publish data to prometheus
  - When used with Thanos, handles downsampled data better than graphite