



# CSCS Tier2 Status LHCb

*Roland Bernet*

*Universität Zürich*



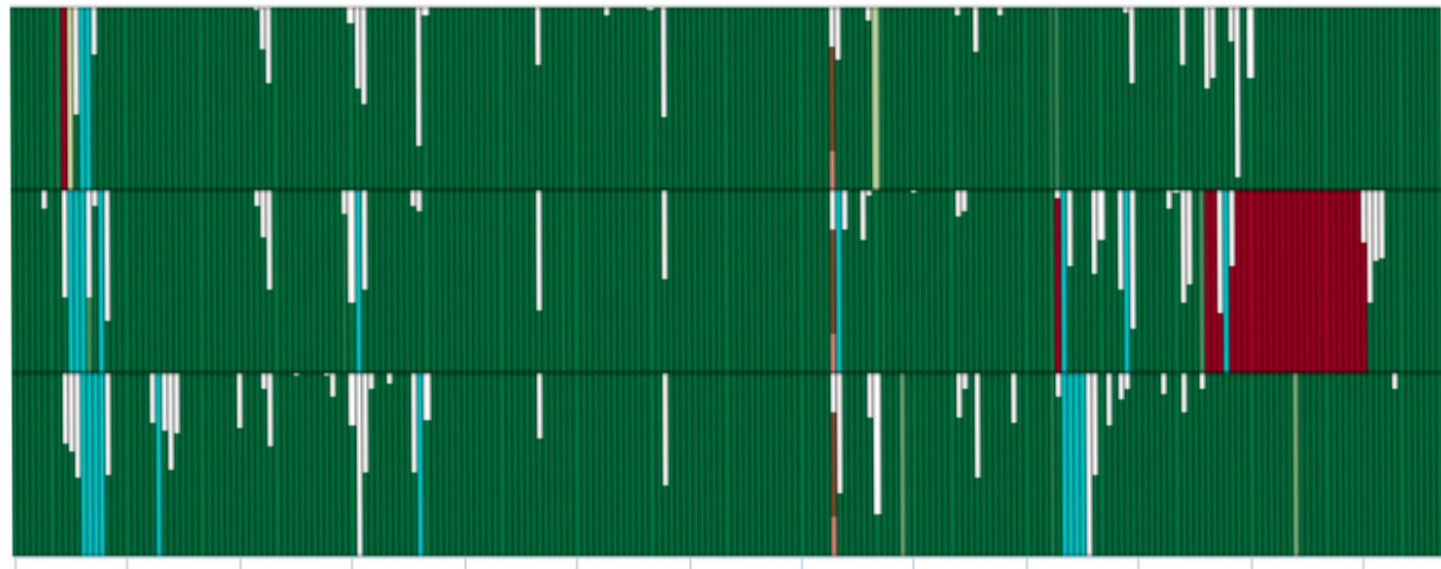
- Status: Generally smooth running for both clusters (Phoenix and Piz Daint).  
Piz Daint has however still slightly higher failure rate for pilots and jobs than Phoenix.
- We had a problem with the queue length definitions on Piz Daint which resulted in jobs getting killed. This got fixed end of November.
- Tickets: In general I do not see CSCS tickets as I am not involved. Tickets are dealt with centrally. I get only involved, when CSCS and LHCb cannot solve them or something has to be discussed and/or a ticket is not the right channel.
- Monitoring: CSCS monitoring:  
The new monitoring plots for Phoenix and Piz Daint are a big improvement and are sufficient for LHCb. Unfortunately you need to tunnel through ela to access the plots.
- LHCb monitoring:  
We have split the site CSCS into LCG.CSCS.ch (Phoenix) and LCG.CSCS-HPC.ch (Piz Daint) and have now the same monitoring for both clusters and we can therefore create the same type of plots.
- :



SAM Availability: 01/06/2018 - 15/01/2019

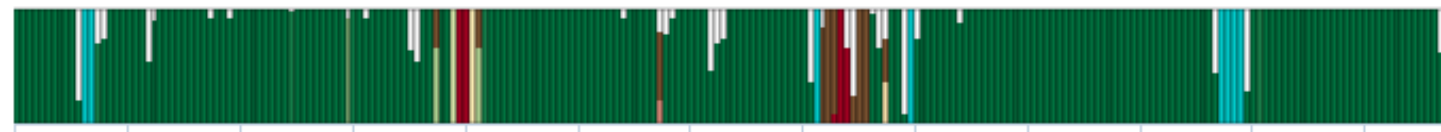
From 2018/06/01 to 2019/01/15

Phoenix ARCs



From 2018/06/01 to 2019/01/15

Piz Daint ARC

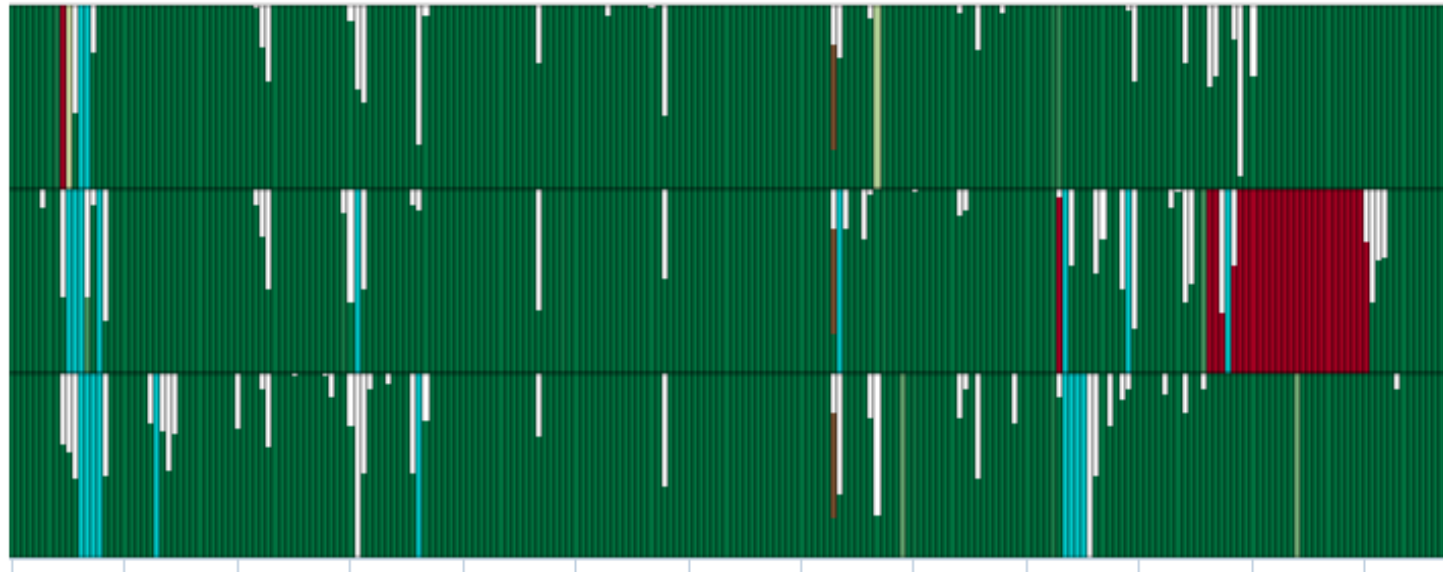




SAM Reliability: 01/06/2018 - 15/01/2019

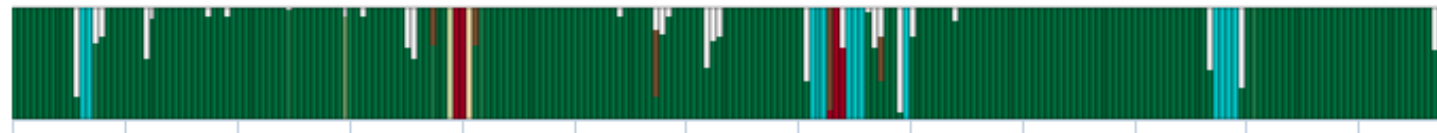
From 2018/06/01 to 2019/01/15

Phoenix ARCs



From 2018/06/01 to 2019/01/15

Piz Daint ARC



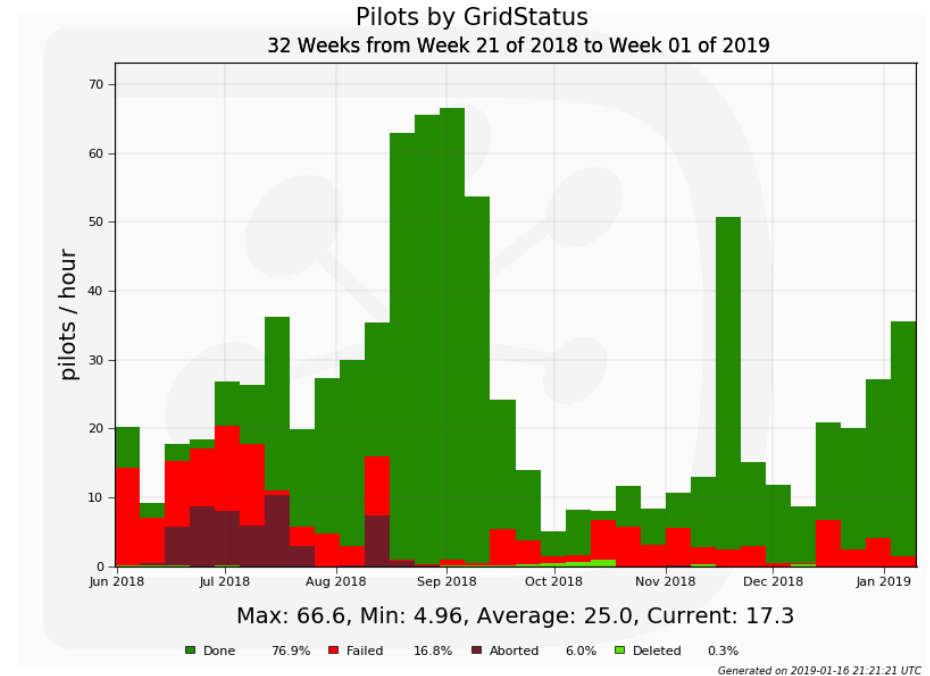
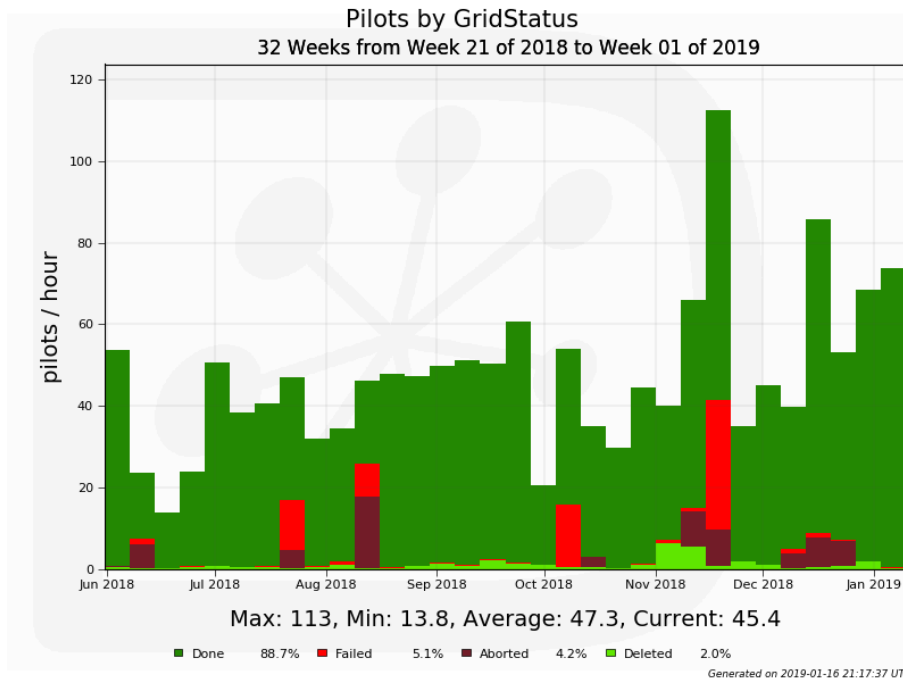


Pilot Efficiency:

01/06/2018 - 15/01/2019

Phoenix

Piz Daint



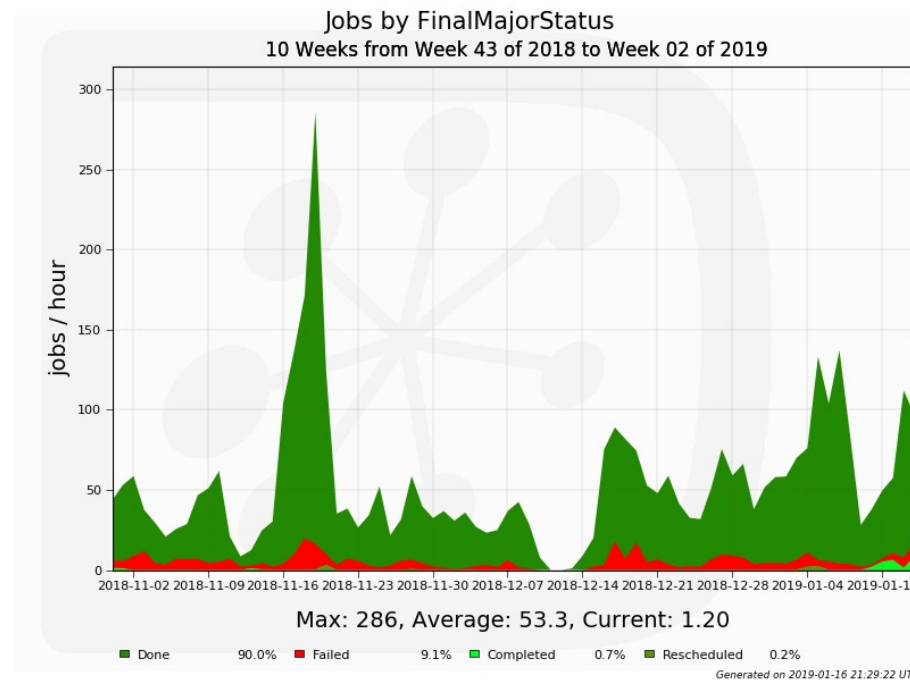
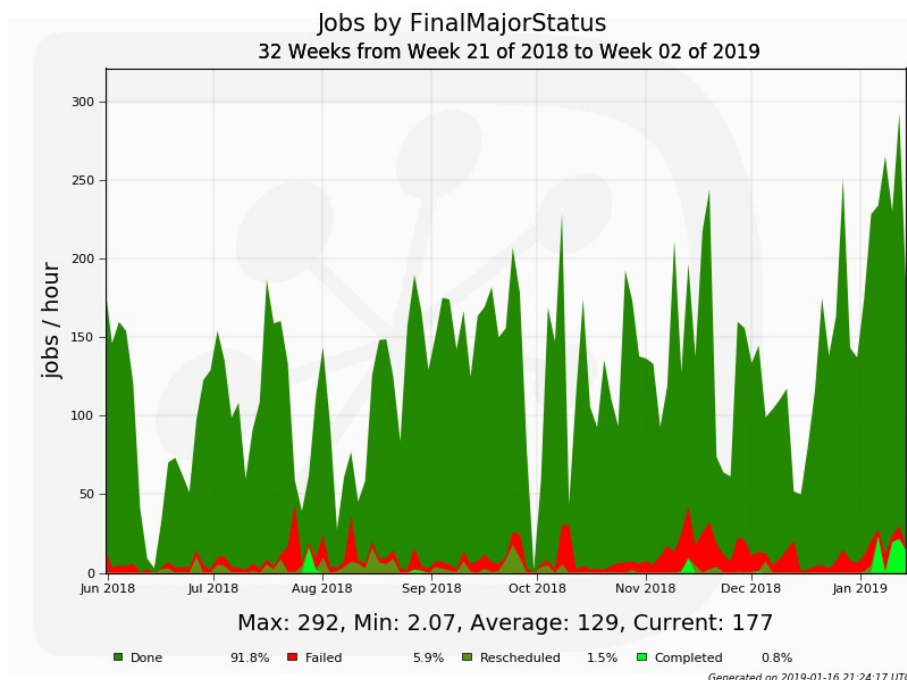


Job Efficiency:

01/06/2018 - 15/01/2019

Phoenix

Piz Daint



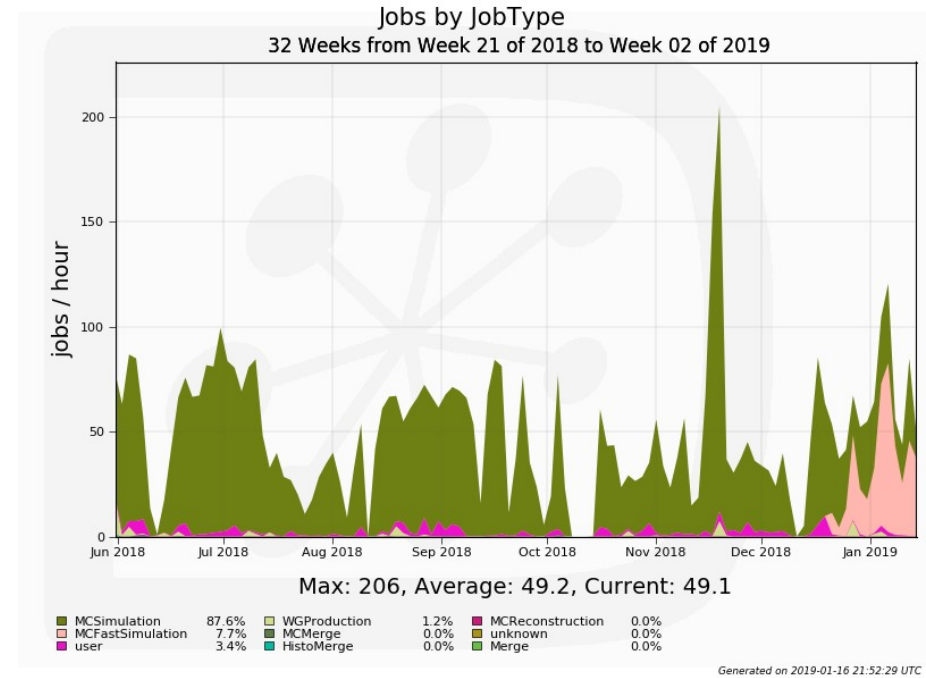
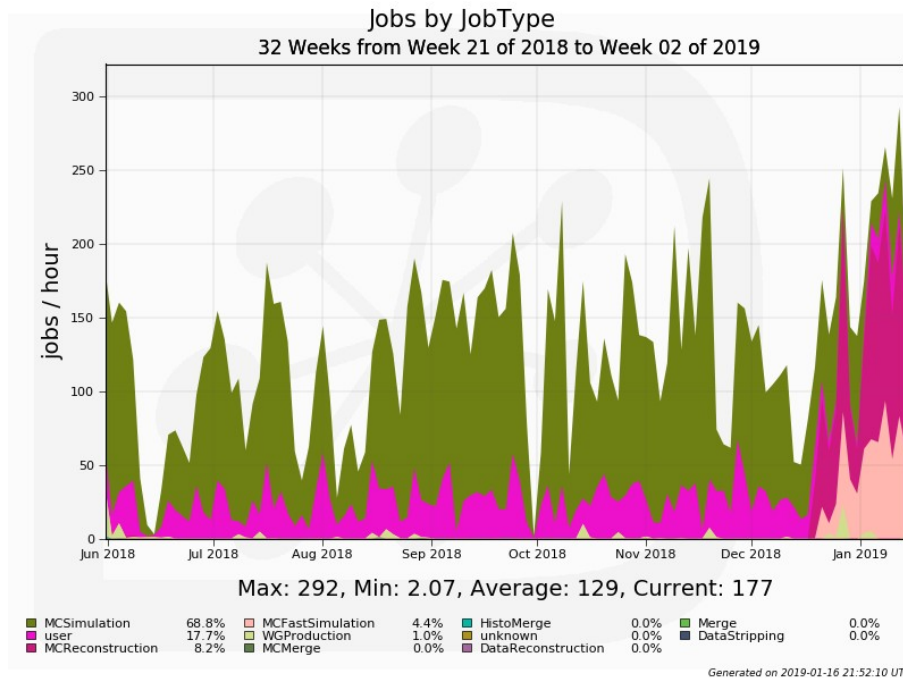


Job Types:

01/06/2018 - 15/01/2019

Phoenix

Piz Daint



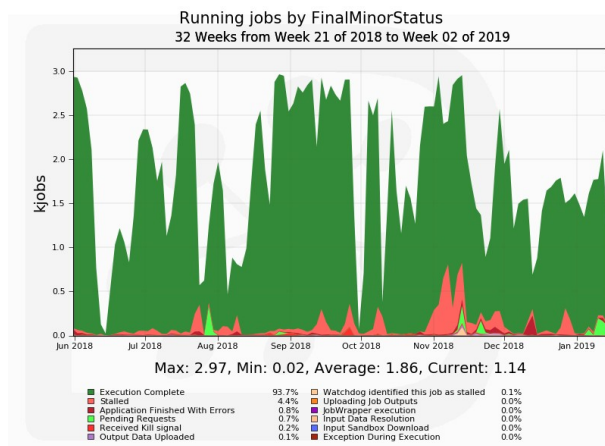




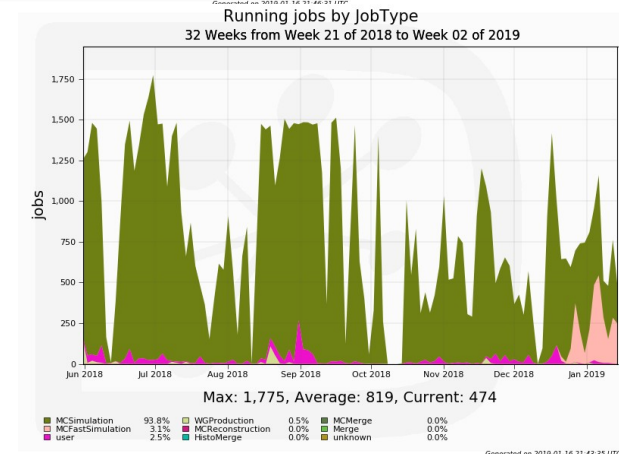
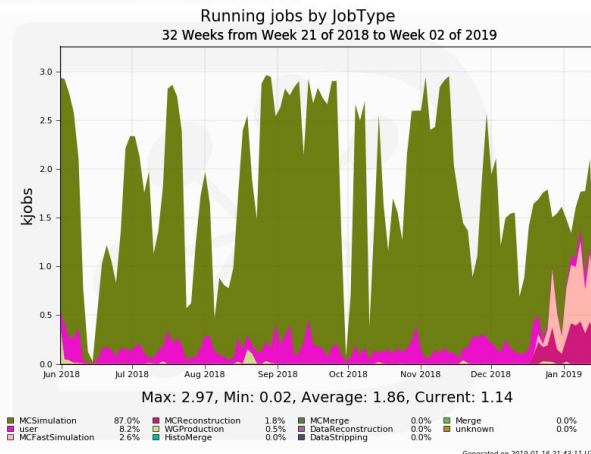
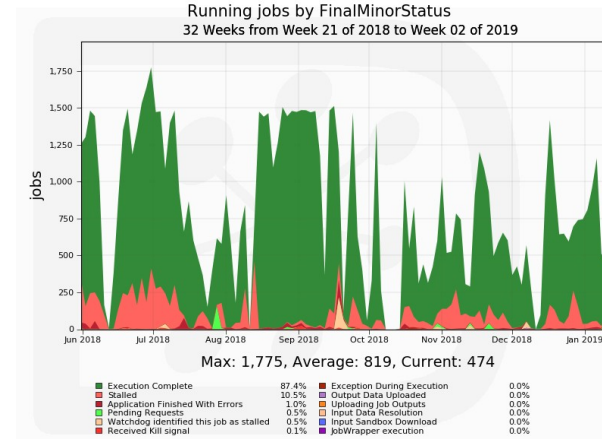
Running Job:

01/06/2018 - 15/01/2019

## Phoenix



## Piz Daint



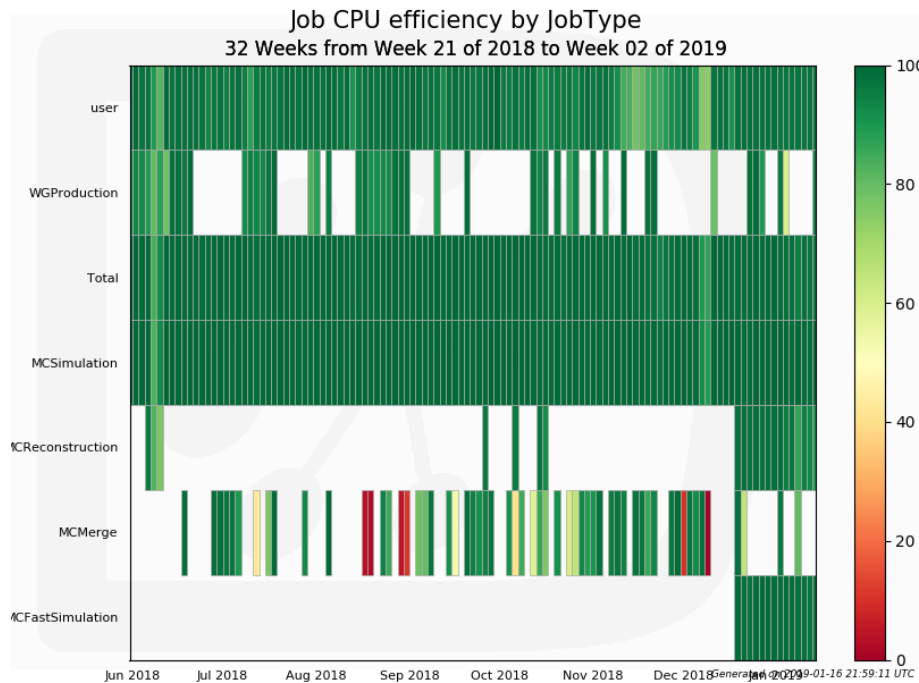




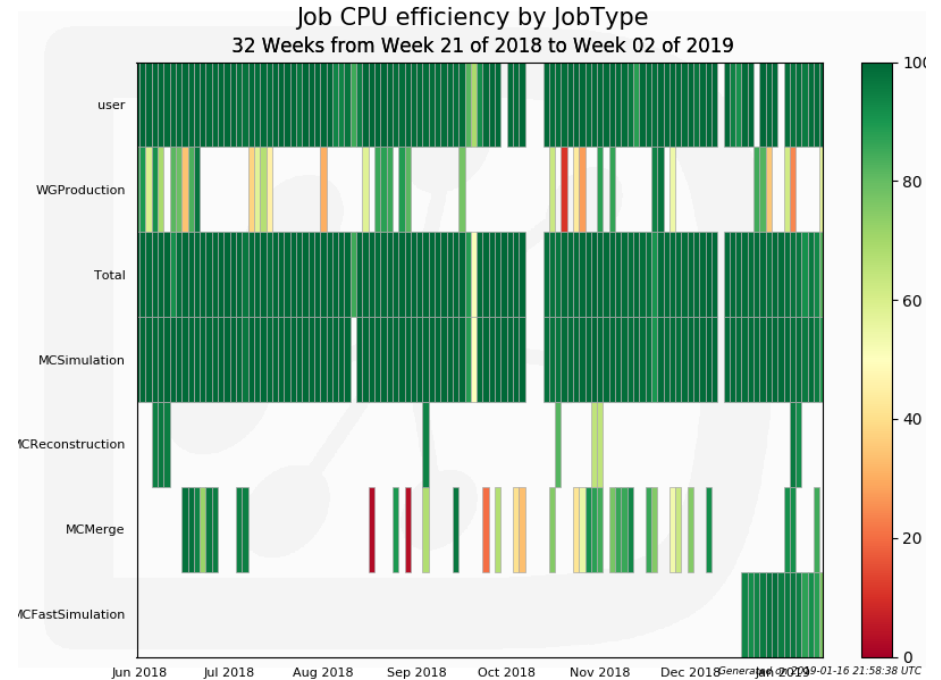
CPU Efficiency:

01/06/2018 - 15/01/2019

Phoenix



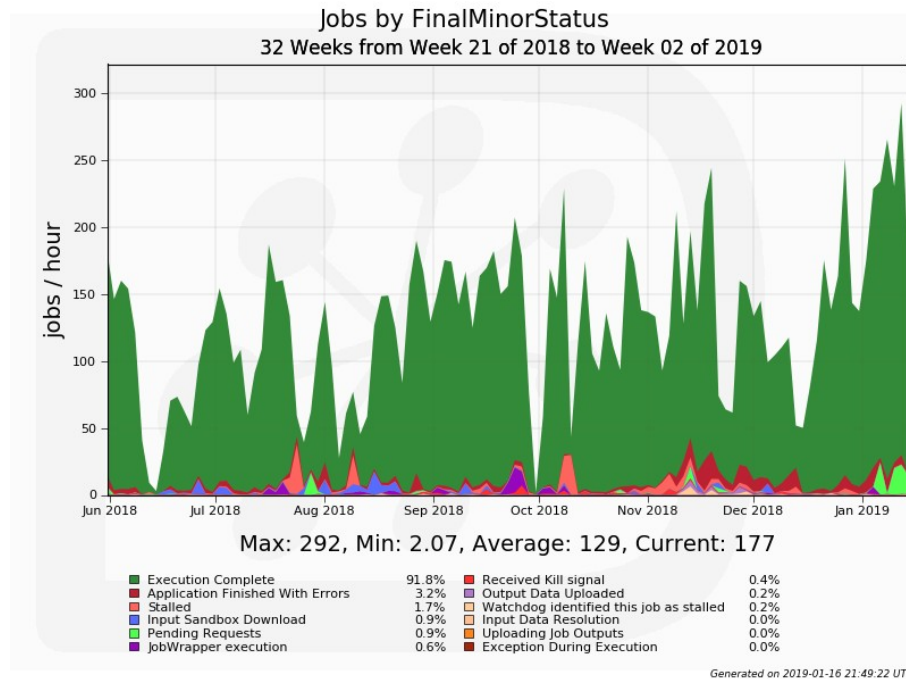
Piz Daint



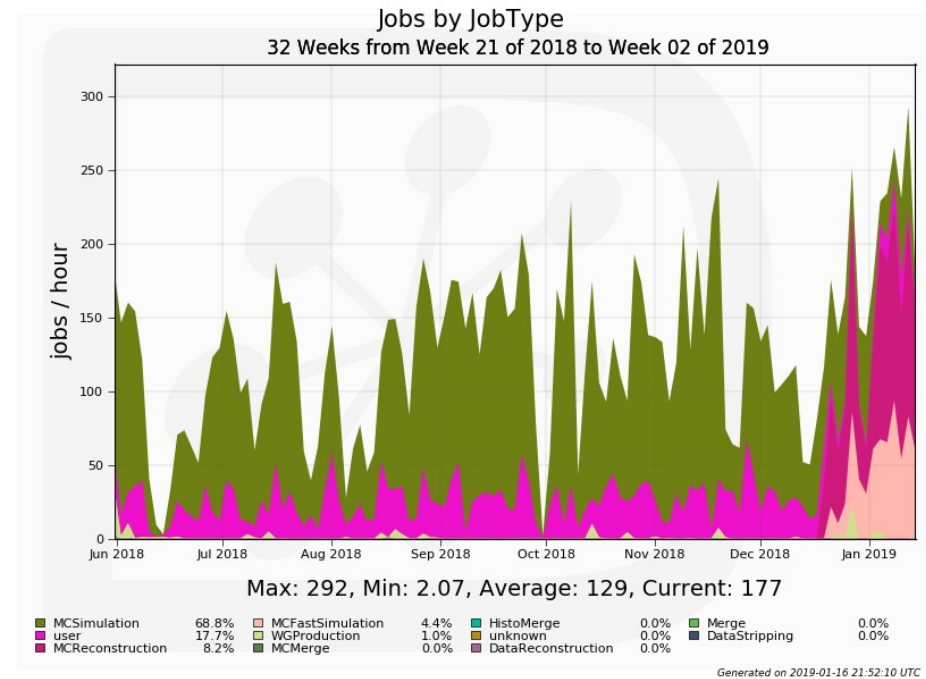


## Phoenix Performance:

### Job Statistic on Phoenix



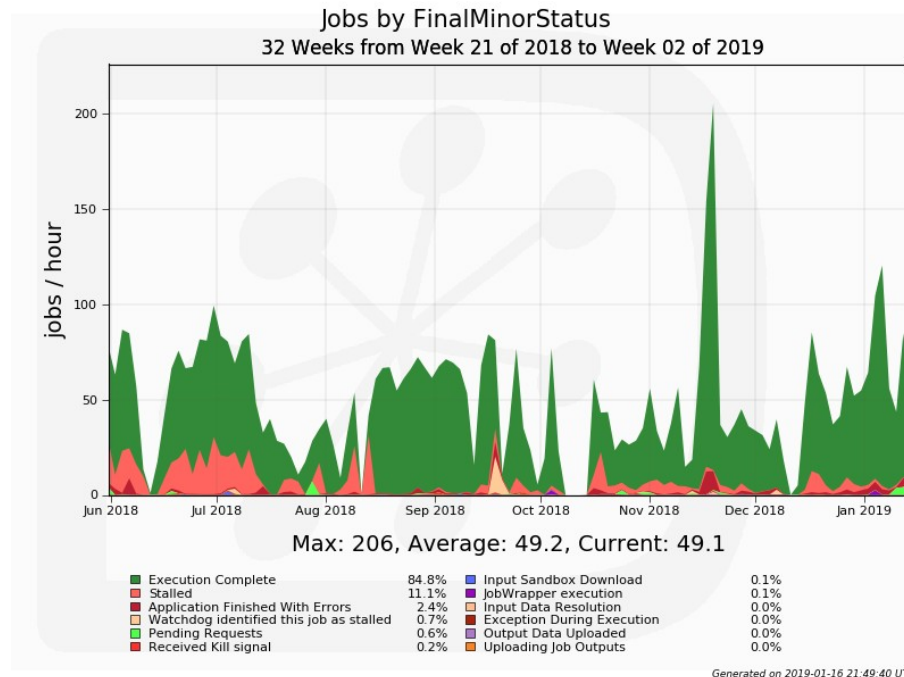
### Job Types on Phoenix



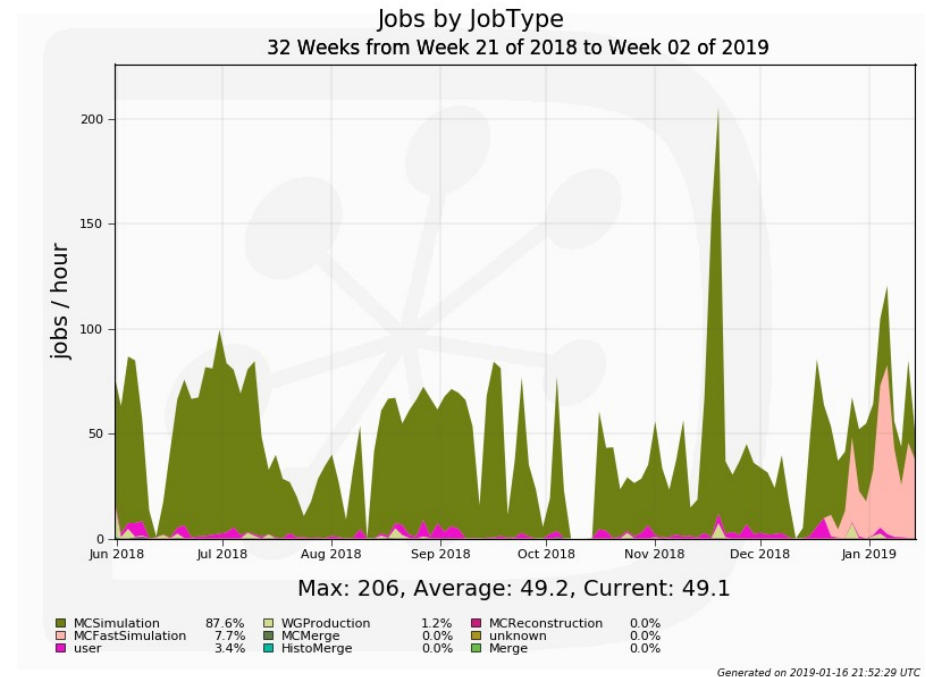


## Piz Daint Performance:

### Job Statistic on Piz Daint



### Job Types on Piz Daint





# UZH - Tier3 Status LHCb

since 01/11/2018  
( fix of queue length problem )

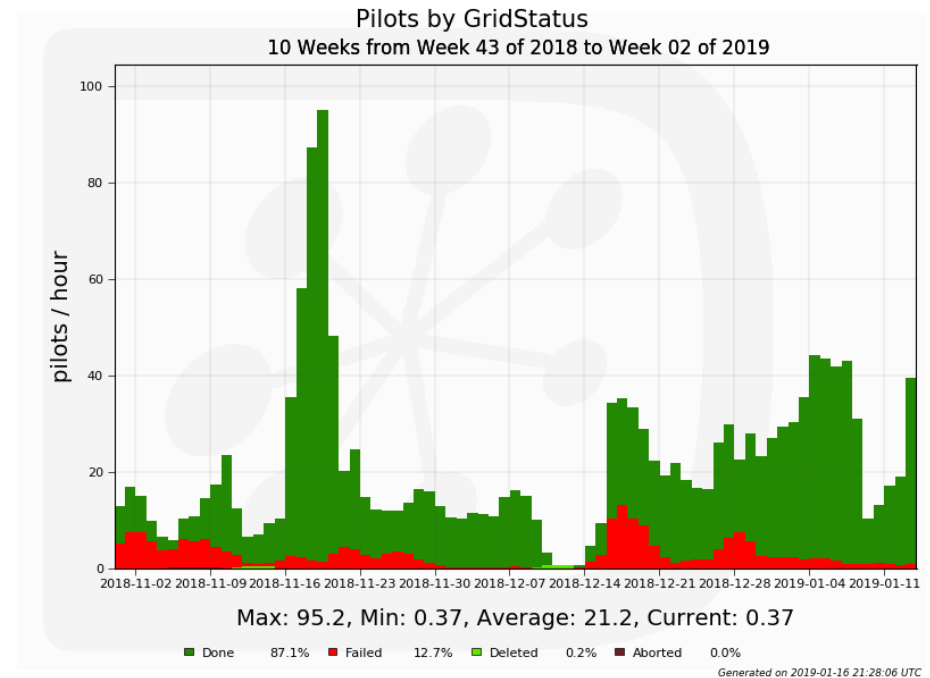
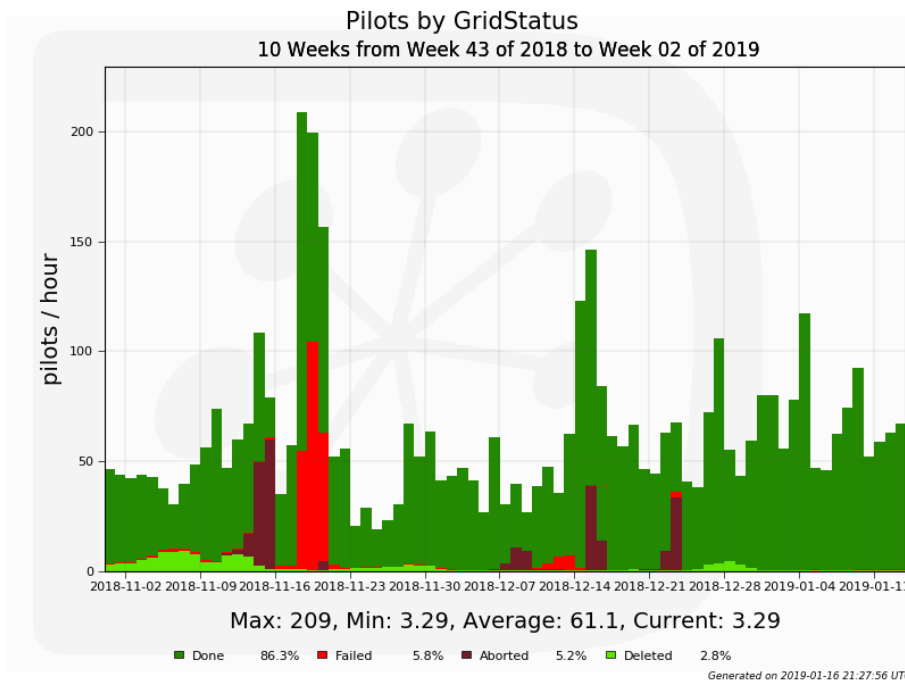


Pilot Efficiency:

01/11/2018 - 15/01/2019

Phoenix

Piz Daint



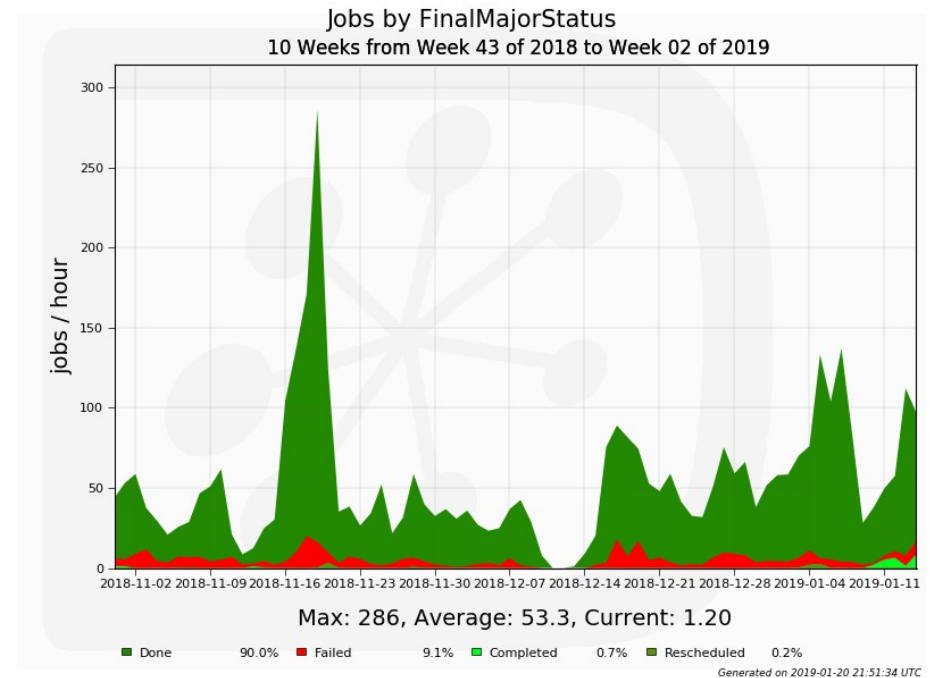
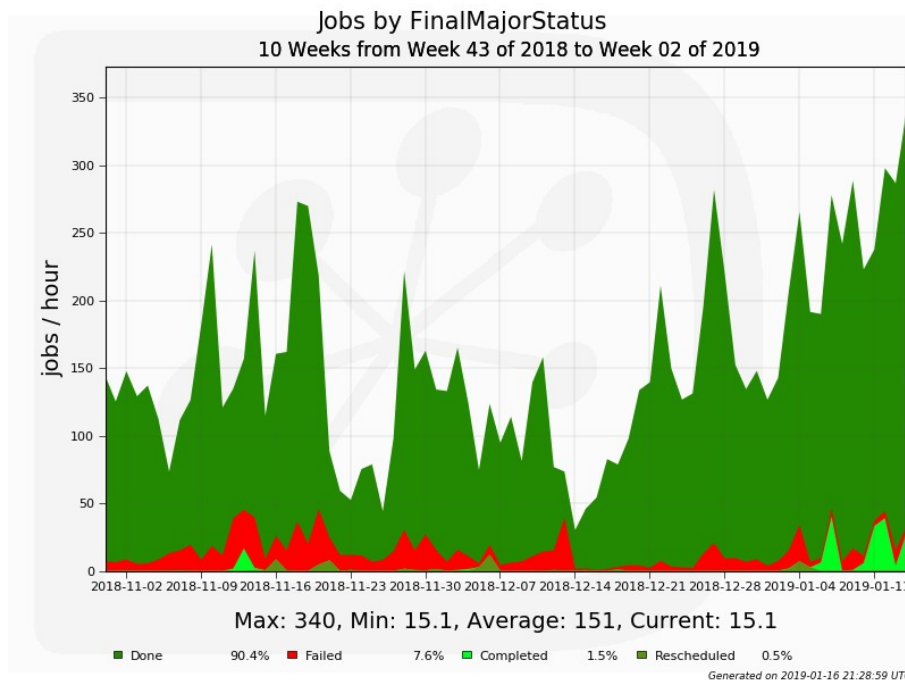


Job Efficiency:

01/11/2018 - 15/01/2019

Phoenix

Piz Daint



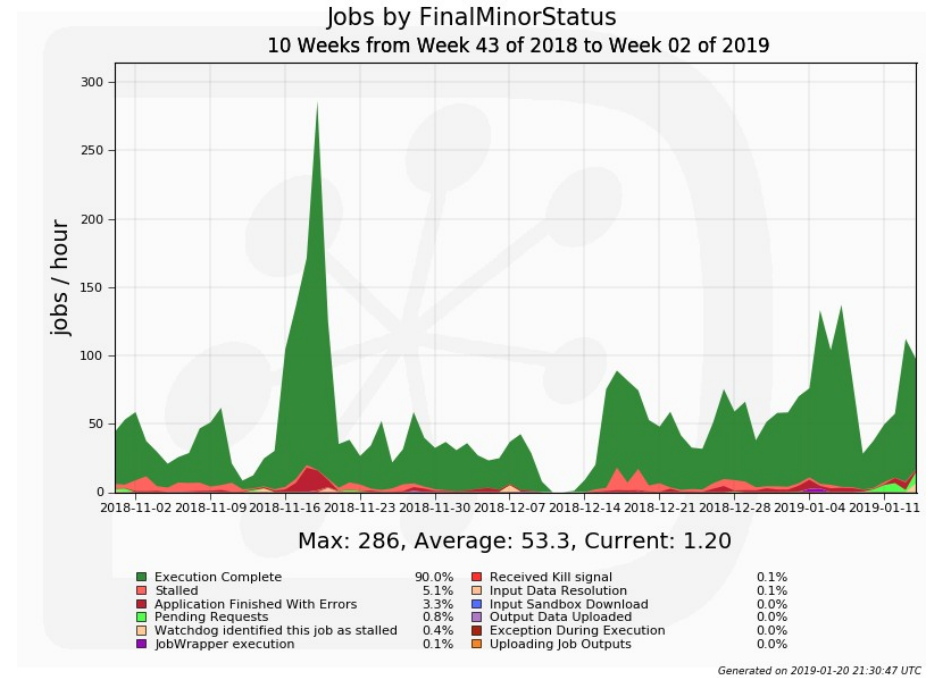
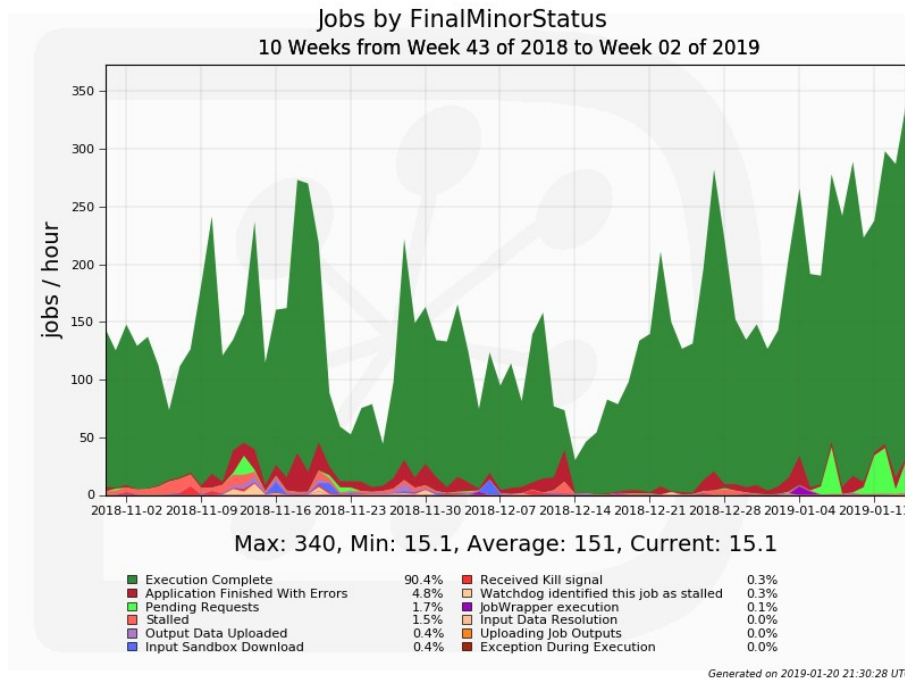


Job Performance:

01/11/2018 - 15/01/2019

Phoenix

Piz Daint







## Conclusions

- We reduced the fraction of stalled jobs on Piz Daint. it is now closer to the level on Phoenix, but still some percentage higher. Also the job failure rate is still a bit higher on Piz Daint.
- Data access on Piz Daint seems to be less efficient than on Phoenix. The job success rate however is similar.
- The changes to the queue length in December do not seem to have a big effect for LHCb jobs running at CSCS. As LHCb started to run new workflows at around same time, it is hard to compare the numbers.  
As it did not get noticeably worse, it seems to work fine for LHCb.

**Thanks for all the work!**



# UZH - Tier3 Status LHCb



Status: LHCb Zürich maintains a local simulation and analysis cluster, which is integrated into the institute Linux cluster. We are not part of WLCG but are part of the LHCb DIRAC framework, which allows us to run LHCb Grid jobs on idle CPUs.

Hardware: - 180 CPU cores (ca. 3000 HS06)  
- 250 TB disk space

Development: Recently we started to use the UZH ScienceCloud, an OpenStack multi-purpose compute and storage infrastructure, for our needs. Instead of replacing old hardware in our cluster, we are using CPUs in the ScienceCloud running our own worker node images. This reduces the maintenance burden for us and should overall be cheaper for everybody. Currently around 40% of the CPU power is delivered by the ScienceCloud. We started to migrate the system scratch area to the ScienceCloud.

Usage: week 21-28 September 2017:  
dirac: LHCb LCG jobs  
others: local user jobs

