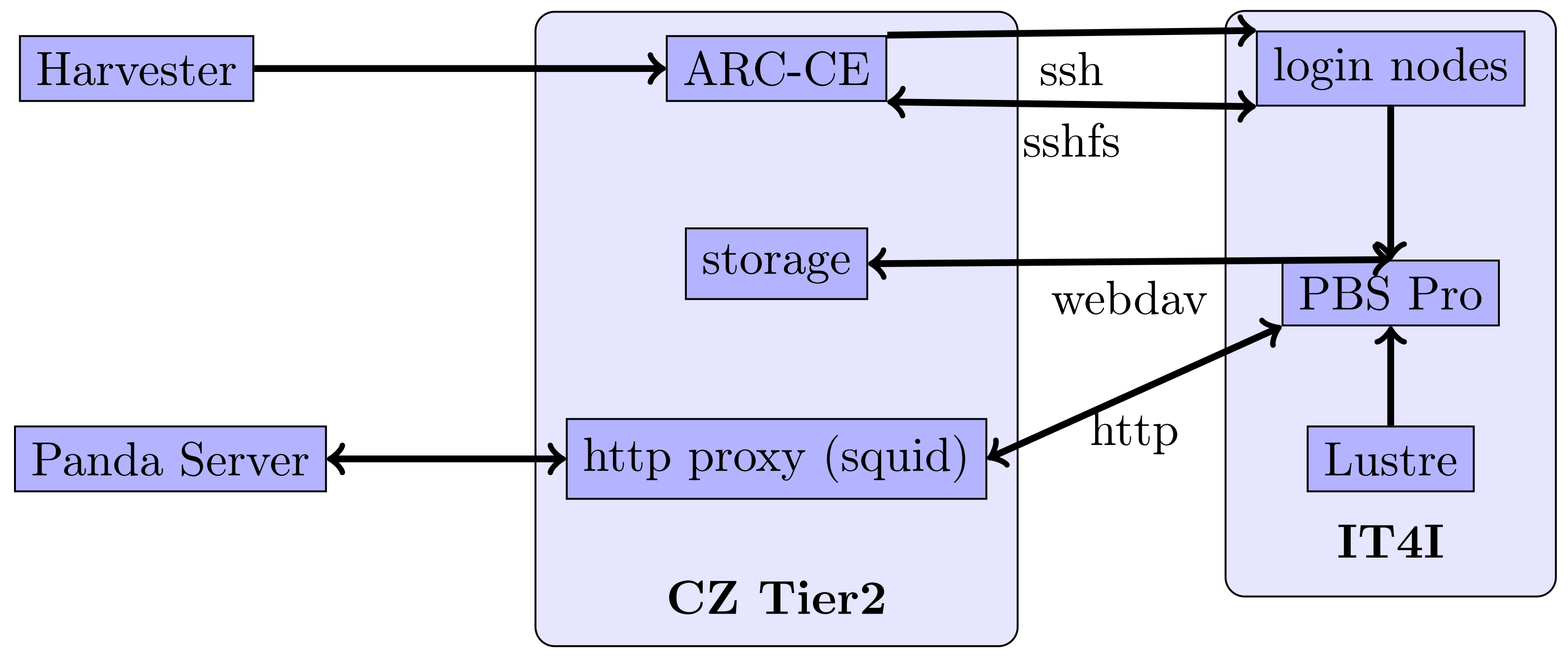
ATLAS job submission systems for the IT4Innovation

M. Svatoš, J. Chudoba, P. Vokáč

5th Users Conference of IT4Innovations

9.-10.11.2021

Submission system using pull model

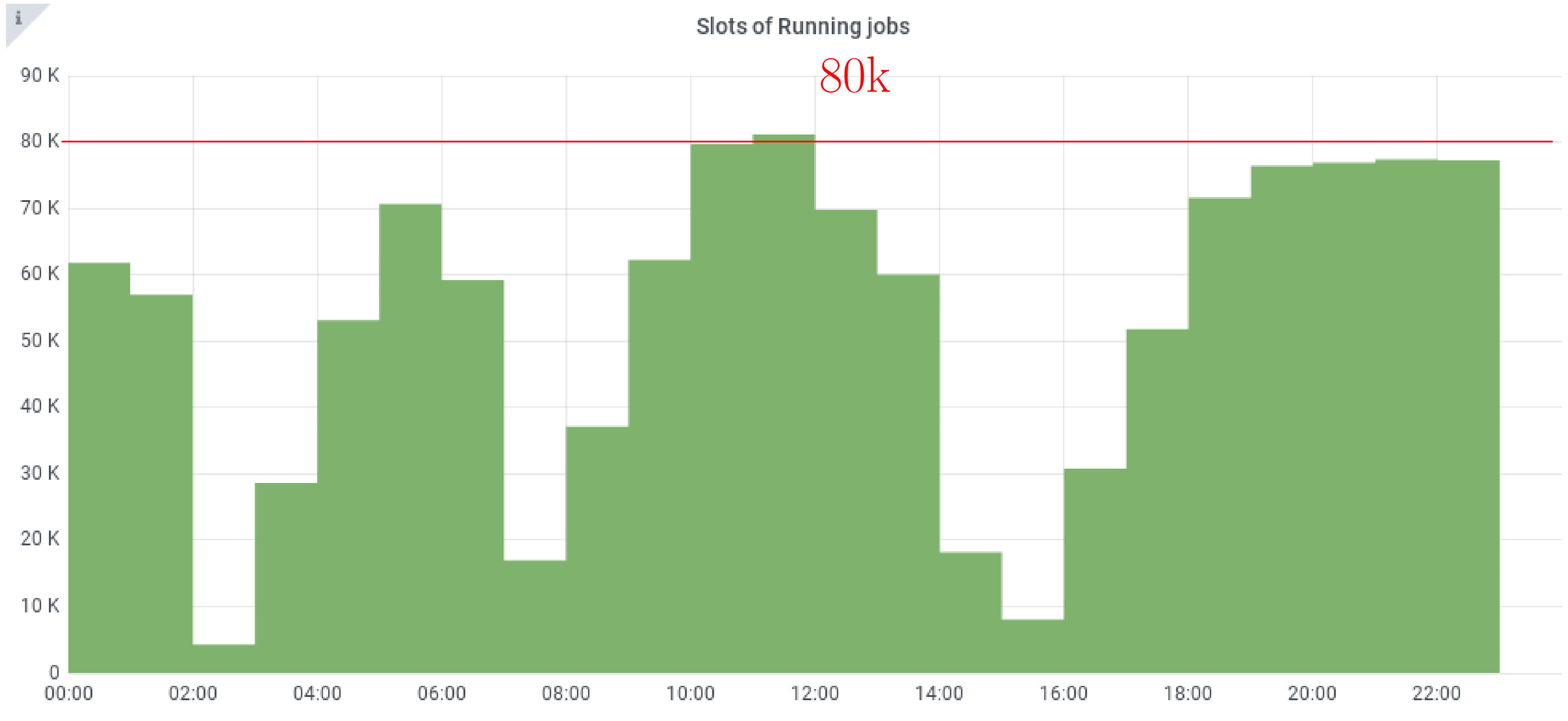


Used on Barbora since June 2021 and on Karolina since August 2021.

- the ARC-CE receives a pilot job, translates the job description into script that can be run in batch system, puts necessary files into a folder shared with the HPC via sshfs, and submits the job via ssh connection to a login node
- when the job starts, pilot contacts panda server through http proxy (CZ Tier2 squid) to receive payload job (as http is one of few open ports)
- if it receives payload job, it gets input file from CZ Tier2 storage via webdav (in modified rucio container)
- then it starts the calculation (in software container)
- when the payload job finishes, it sends outputs to CZ Tier2 storage via webdav (in modified rucio container)
- when this is finished, pilot will request another payload job (if it can expect that the batch queue setting would allow it to finish)

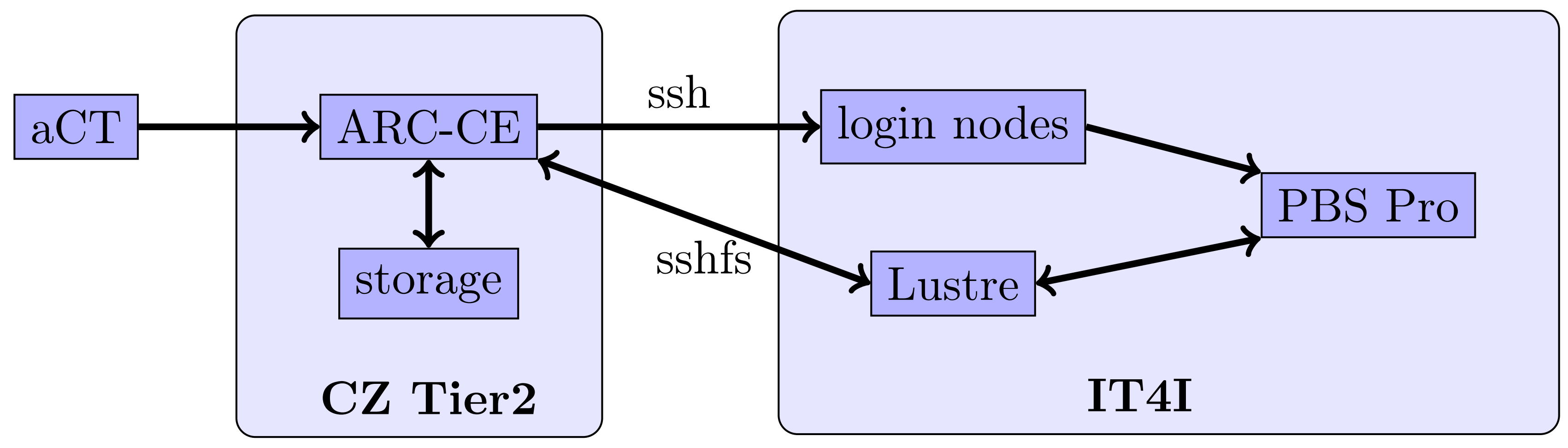
Filling Karolina

At maximum, this system was able to utilize over 81k of Karolina's 92k cores.



running jobs slots on 23.8.2021

Submission system using push model



Used on Anselm and Salomon. On Barbora, it was used until June 2021.

- an ARC-CE receives job description from the ARC Control Tower (aCT) submission system
- it downloads input files from storage storage and put all into folders that are shared between ARC-CE and dedicated scratch space via sshfs - either job session directory or input file cache
- it translates aCT job description into script that can be run in batch system and submits it via ssh connection to a login node
- running jobs use software located also on the scratch space
- ARC-CE obtains the output of finished job via sshfs and uploads it to CZ Tier2 storage

Submission systems comparison

push model

+ much simpler (it needs only the ARC-CE machine)

— only one payload is run per job - when it is finished, the batch slot is lost

— all files (input files, scripts, etc.) go through the same sshfs connection

* to achieve reasonable frequency of starting jobs, sizeable cache (several TB) of input files is needed on the Lustre

pull model

+ batch slot could be kept almost as long as batch queue limit allows

+ there is no need for an input file cache

+ data transfers happen independently for each job

— requires http proxy machine in addition

— needs special container for moving files which enforces transfers via http

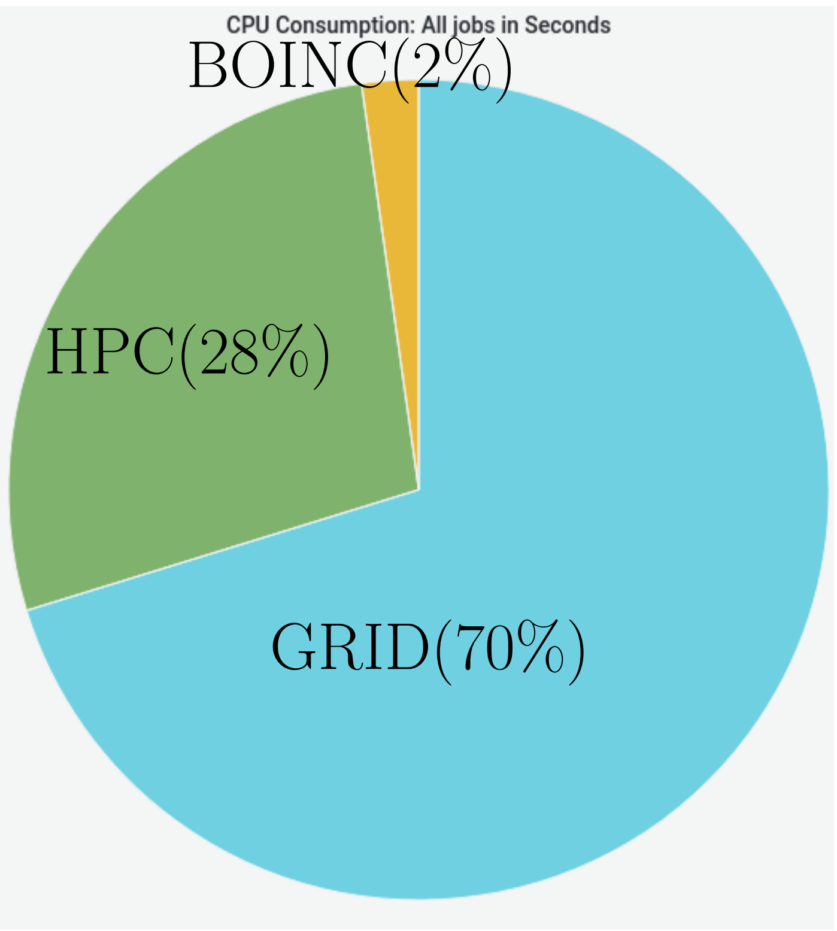
Containerization

ATLAS can use (and often uses) Singularity containers to run its workloads. On HPCs of IT4Innovation, two kinds of containers are used

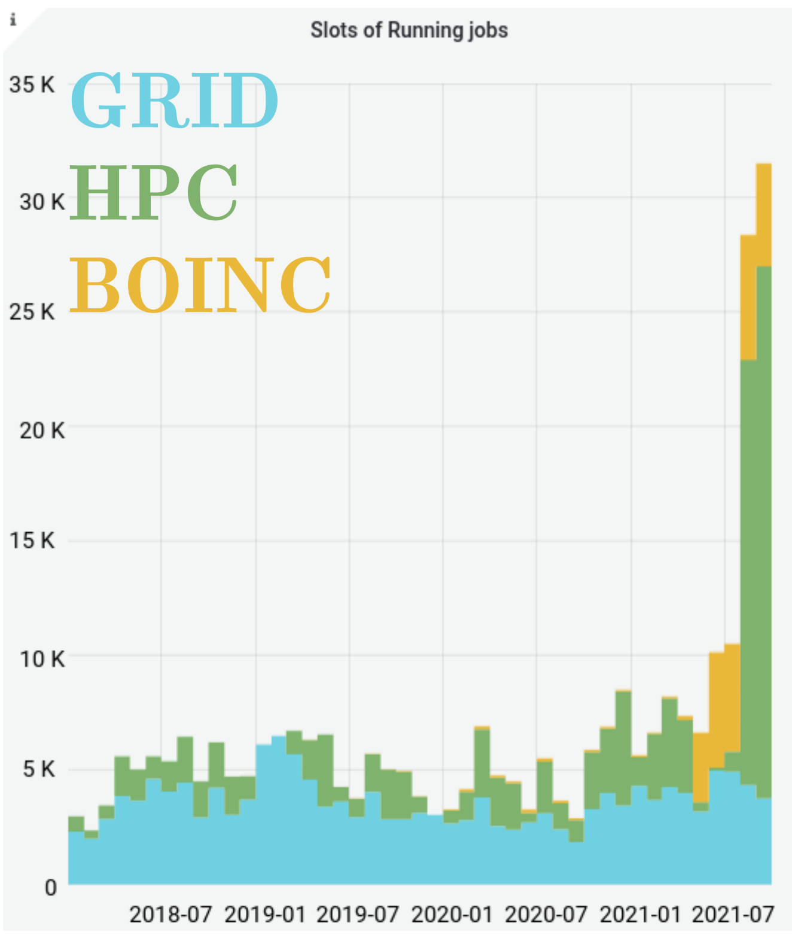
- software container: it contains all necessary files to run ATLAS calculations
- rucio container: it is used for stage-in (download of input data) and stage-out (upload of output data) in the pull model

Performance

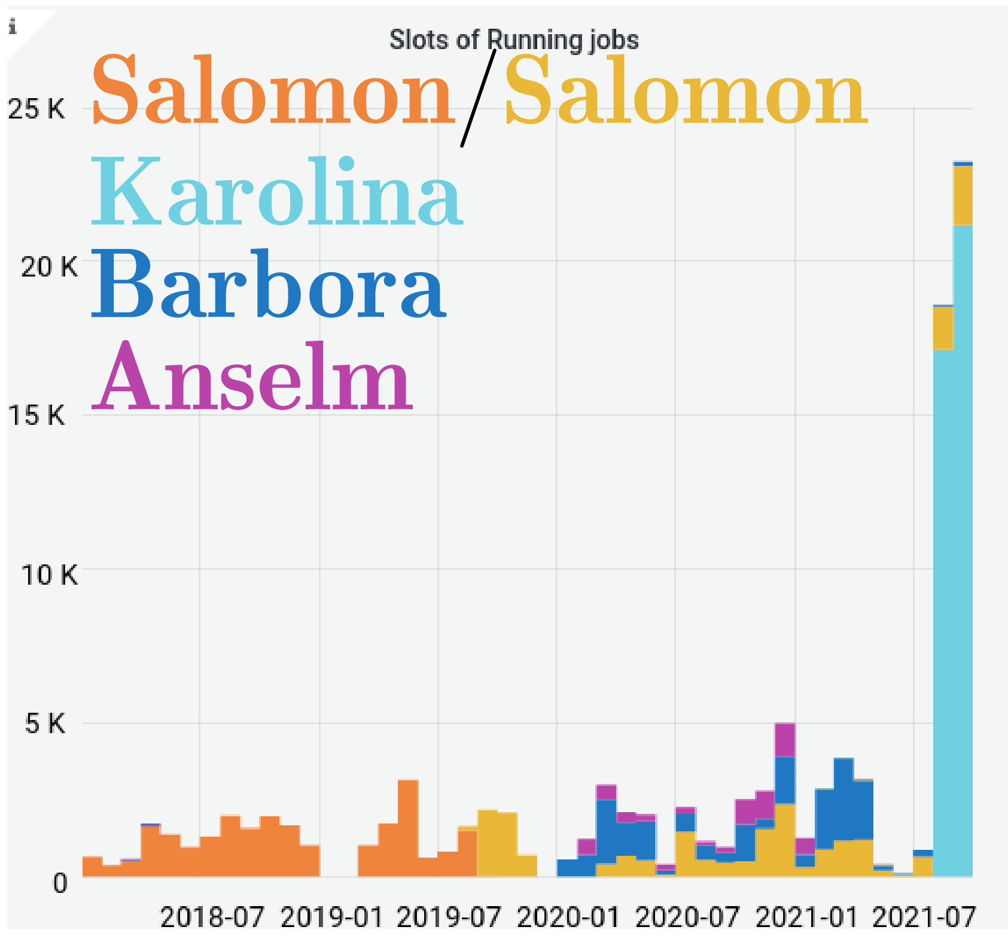
CPU consumption (in seconds) of all CZ Tier2 resources



Monthly average of number of cores used on all CZ Tier2 resources



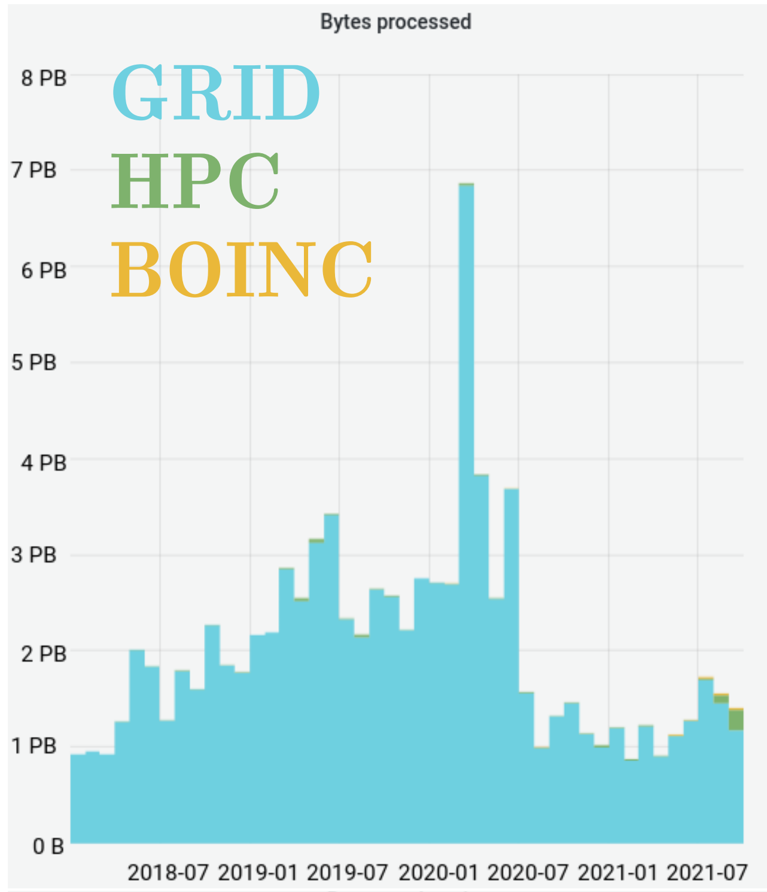
Monthly average of number of cores used on IT4I HPCs

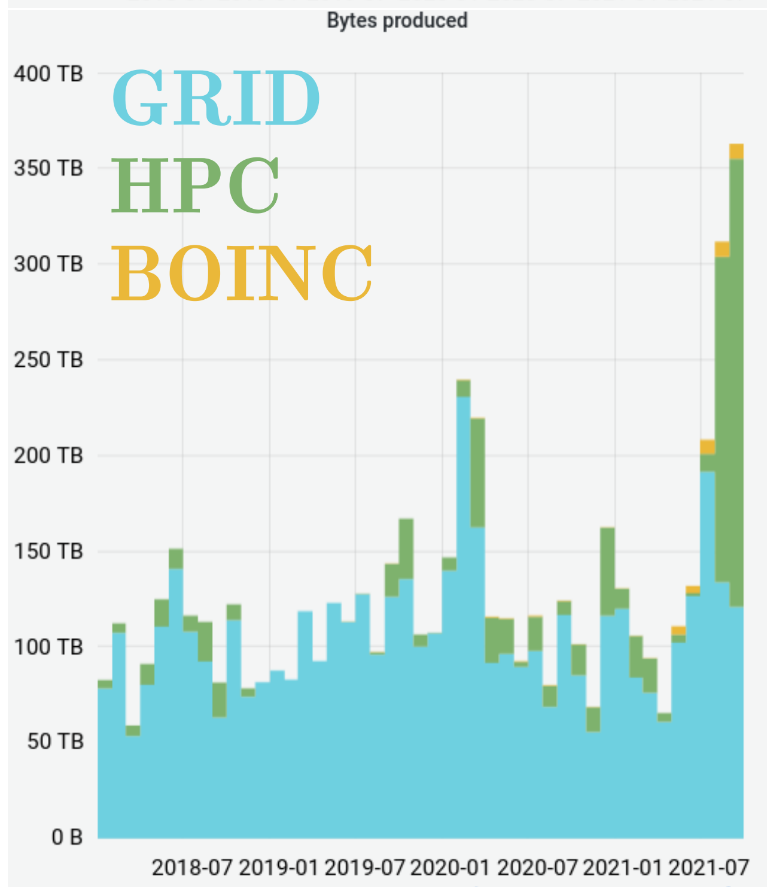


Acknowledgement

This work and computing resources at FZU were co-financed by projects CERN-C (CZ.02.1.01/0.0/0.0/16013/0001404) and CERN-CD (CZ.02.1.01/0.0/0.0/18_046/0016013) from EU funds and MŠMT and projects CERN-CZ (LM2018104) and LTT17018.

Possible improvements





- storing more of ATLAS software (millions of small files)
  - on dedicated software area
  - using CVMFS installation
- with more software available, more IO demanding jobs would arrive to IT4I which could strain the networking - compare grid and hpc on the left figures
  - monthly sum of bytes processed (i.e. input files) coming to IT4I is only few percent of what is used on grid
  - monthly sum of bytes produced (i.e. output files) are at level similar to grid since Karolina was put into production while using many times more cores