# Updates on usage of the Czech national HPC center

M. Svatoš, J. Chudoba, P. Vokáč

On behalf of the ATLAS Collaboration

vCHEP 2021, 17-21.5.2021

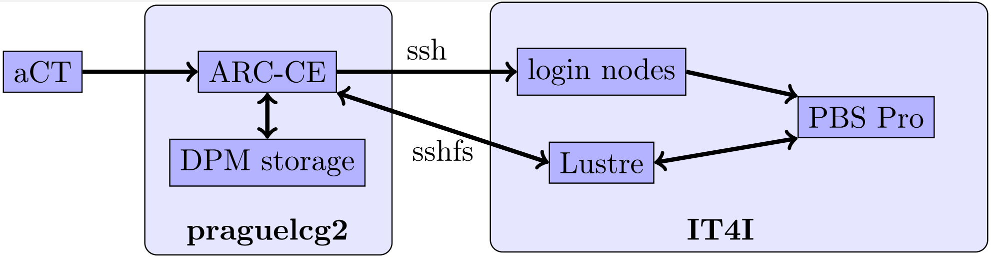
# Introduction

The ATLAS distributed computing is opportunistically using resources of the Czech national HPC center IT4Innovations through Czech Tier2 praguelcg2:

- Salomon (since December 2017) - to be decommissioned in 2021
  - ∗ 24 cores (Intel Xeon E5)
  - ∗ 128 GB of RAM
- Barbora (since January 2020)
  - ∗ 36 cores (Intel Cascade Lake 6240)
  - ∗ 192 GB of RAM
- Anselm (February 2020 to January 2021) - now decommissioned
  - ∗ 16 cores (Intel Sandy Bridge)
  - ∗ 64 GB of RAM

A new machine, Karolina (128 cores, 256 GB of RAM), is in preparation.

# Job submission system (push model)



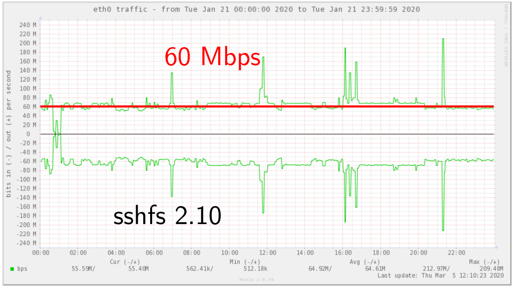
- the ARC Control Tower (aCT) submits job description to one of the ARC-CE machines
- the ARC-CE processes the description and submits it to the HPC via ssh connection to login node
- job's auxiliary files are shared between ARC-CE and Lustre storage via sshfs connection
- software used by running jobs is located on the Lustre storage
- output of finished job is transferred from the Lustre storage back to the ARC-CE machine via sshfs

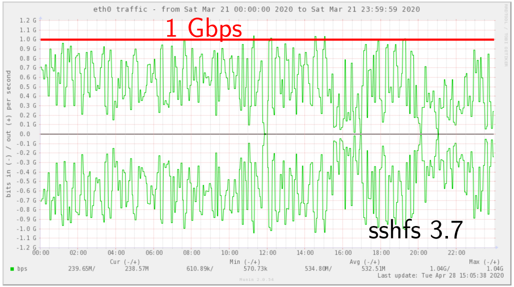
Push model is suitable for restrictive environment where only ssh connection though login nodes is available ,i.e. not suitable for data intensive workloads (but support for those would be appreciated)

# Migration to ARC-CE version 6

- when this system was set up, the available ARC-CE version was 5
- later, version 6 was released with improved reliability and scalability, new systems for runtime environment scripts and accounting, etc..
  - solved our issues with accounting records lost during synchronization to APEL
- all ARC-CEs were migrated to version 6 during June/July of 2020.

# Sshfs

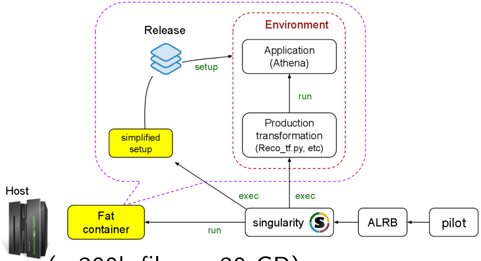




- the connection via sshfs is a bottleneck of this system
  - network saturation at about 60 Mbps on CentOS7 default version of sshfs (2.10)
- sshfs version 3.7 was released in January 2020
  - it introduced `max_conns` option which enables the use of multiple connections
  - with 10 connections, there is no saturation (with peaks over 1 Gbps)

# Containerization

- ATLAS can run its jobs inside of Singularity containers since 2019
  - CVMFS access is required
- the CVMFS is not installed on HPCs (but it would be nice if this is something considered for future HPCs)
- singularity is installed on HPCs



fat container

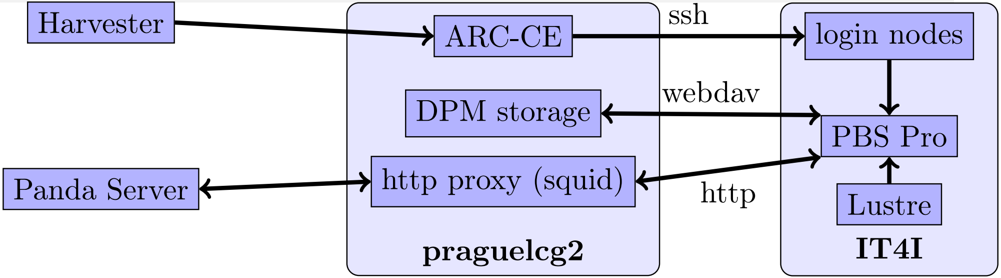
- all the necessary software and condition data of one release (∼300k files, ∼30 GB)
- advantages
  - number of files
    - ∗ without the fat container, part of the CVMFS needed to be located on the HPC's storage (12 million files)
    - ∗ with fat container, about 10 million files could be removed
  - hard to reproduce and debug timeouts to squid at praguelcg2
    - ∗ the fat containers do not need to contact a squid

# Long jobs

Newly prepared HPC of IT4Innovation (Karolina; to be installed in 2021) is supposed to have worker nodes with 128 cores

- on 128 cores, ATLAS simulation jobs would become rather short and opportunistic usage would decrease
- there are several ways to increase the length:
  - to have specific tasks with more events per job
    - **pros:** no changes would be needed in the site setup
    - **cons:** this costs time of production team experts
  - run several jobs in parallel using harvester
    - **pros:** execution can be shaped in any way we need
    - **cons:** harvester is not CE replacement (it does not provide necessary functionalities like reporting the accounting data into the APEL)
  - run several jobs run within one pilot job
    - ∗ this is basically a pull model which is running on the grid (the push model is described on slide 3)
    - ∗ on closed environment like HPC, it is tricky to make it work - jobs cannot contact panda servers to get payload, cannot stage-in and stage-out
    - ∗ for pull model, HPC's worker nodes have to have outbound connectivity

# Long jobs



- the ARC-CE receives a pilot and submits it to the HPC via ssh connection to a login node
- when the job starts in the batch system, pilot contacts panda server through http proxy (praguelcg2 squid) to receive payload
- if it receives payload, it gets input file from DPM on praguelcg2 via webdav (in modified rucio container)
- when the payload finishes, it sends outputs to DPM on praguelcg2 via webdav (in modified rucio container)
- if this is finished within job length limit (defined in CRIC), pilot will request another payload

**Acknowledgement:** Thanks to Rod Walker et all for the help with this setup

# Long jobs

Timing:

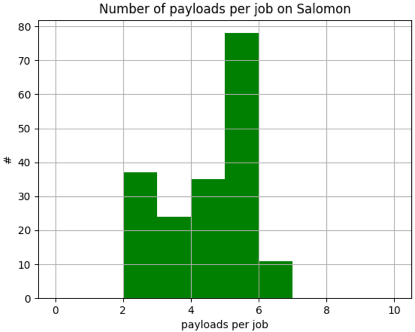
- Analysis of job phases timing:
  - file transfers (stage-in, stage-out): few seconds to few minutes
  - single-core part of the simulation: ∼ two minutes
  - multi-core part of the simulation: this can vary significantly
    - example: ∼ two hours on 24-core nodes → ∼ twenty minutes on 128-core nodes
  - merge: ∼ two minutes
- unless the multi-core part would be only few minutes long, the CPU efficiency might be acceptable on 128 cores

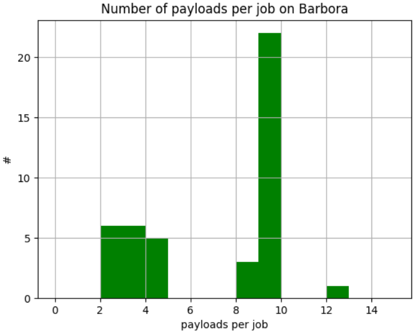
Issues:

- proxy length issues
- no jobs in panda causing empty pilots

# Long jobs

Number of payloads per job:



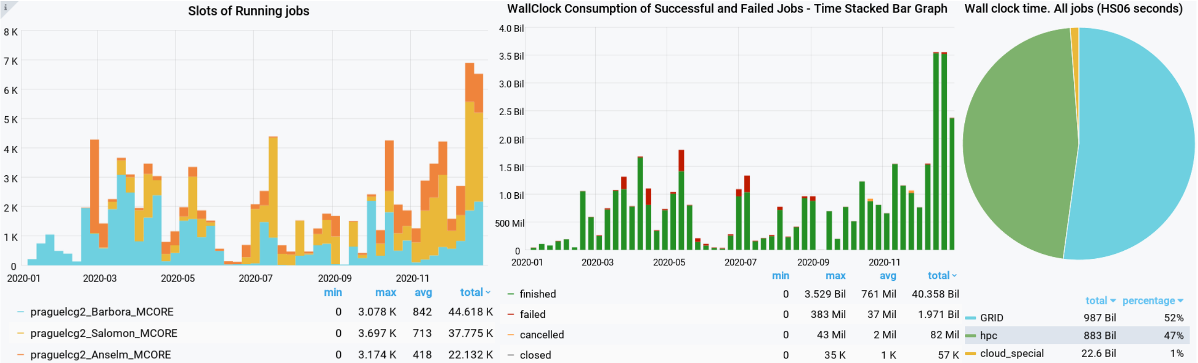


- plots ignore jobs with

  0 payload - empty pilot
  1 payload - failing stage-in/stage-out during testing

- when using allowed time of the opportunistic queue fully, we can run up to ∼5 payloads in one job on Salomon cluster and ∼10 payloads in one job on Barbora cluster

# Performance (2020)



Slots of Running jobs

| | min | max | avg | total |
|---|---|---|---|---|
| — praguelcg2_Barbora_MCORE | 0 | 3.078 K | 842 | 44.618 K |
| — praguelcg2_Salomon_MCORE | 0 | 3.697 K | 713 | 37.775 K |
| — praguelcg2_Anselm_MCORE | 0 | 3.174 K | 418 | 22.132 K |

WallClock Consumption of Successful and Failed Jobs - Time Stacked Bar Graph

| | min | max | avg | total |
|---|---|---|---|---|
| — finished | 0 | 3.529 Bil | 761 Mil | 40.358 Bil |
| — failed | 0 | 383 Mil | 37 Mil | 1.971 Bil |
| — cancelled | 0 | 43 Mil | 2 Mil | 82 Mil |
| — closed | 0 | 35 K | 1 K | 57 K |

Wall clock time. All jobs (HS06 seconds)

| | total | percentage |
|---|---|---|
| — GRID | 987 Bil | 52% |
| — hpc | 883 Bil | 47% |
| — cloud_special | 22.6 Bil | 1% |

- peaks in number of running job slots characteristic for opportunistic nature of the HPC usage
- the failure rate is rather low and comes as short peaks (often coming from infrastructure or central problems)
- the opportunistic resources provided almost half of the wallclock of praguelcg2 in 2020

# Summary and Conclusions

- migration to ARC-CE version 6 fixed problems with synchronization of accounting records to APEL
- with sshfs version 3.7 using 10 connections. there is no network saturation between the ARC-CE and HPCs
- fat containers allowed us to delete 10 million files on each HPC and avoid timeouts to squid
- system which allows running multiple payloads in one job was tested and is prepared for the newest HPC
- the HPCs provided almost half of the wallclock of praguelcg2 in 2020

# Acknowledgement

Computing resources at FZU are co-financed by projects Research infrastructure CERN (CERN-CZ) and OP RDE CERN Computing (CZ.02.1.01/0.0/0.0/16013/0001404) from EU funds and MŠMT. This presentation is supported by project LTT17018.
We gratefully acknowledge National Supercomputing Center IT4Innovations (IT4I) for computing time provided by the Salomon supercomputer.

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research,
Development and Education

MŠMT
MINISTRY OF EDUCATION,
YOUTH AND SPORTS