Using HyperQueue to run multiple ATLAS workloads at IT4Innovations

Michal Svatoš
On behalf of the ATLAS Collaboration

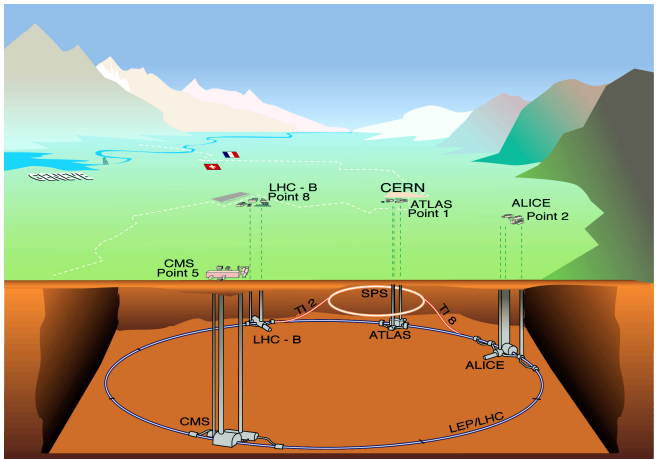
FZÚ, AV ČR

10th Users' Conference of IT4Innovations, 5.-6.11.2026

Content

- LHC
- ATLAS
- WLCG
- CZ-Tier-2
- Hyperqueue
- submission system
- job mixing
- performance in 2026

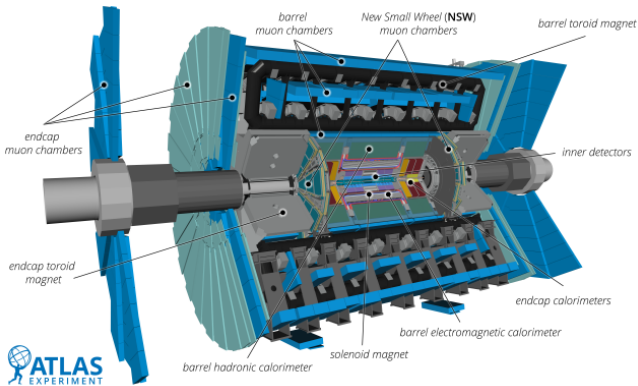
LHC



https://cds.cern.ch/record/842700

- The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator.
- In a tunnel 100 metres underground at the European Organization for Nuclear Research (CERN) on the Franco-Swiss border near Geneva, Switzerland.
- It is a 27-kilometre ring of superconducting magnets.

ATLAS Experiment



https://cds.cern.ch/record/2837191

- ATLAS = A Toroidal LHC ApparatuS
- It is one of two general-purpose detectors at the LHC.
  - 46 m long
  - 25 m high
  - 25 m wide
  - weighting 7000-tonne
- It is currently in long shutdown to prepare for "high luminosity" phase, which is planned to produce 10x more data per year than before.

ATLAS Experiment

- One of its greatest achievements was discovery of Higgs boson, for which the Nobel price was awarded in 2013.

> The Nobel Prize in Physics 2013 was awarded jointly to François Englert and Peter W. Higgs "for the theoretical discovery of a mechanism that contributes to our understanding of the origin of mass of subatomic particles, and which recently was confirmed through the discovery of the predicted fundamental particle, by the ATLAS and CMS experiments at CERN's Large Hadron Collider"
>
> `https://www.nobelprize.org/prizes/physics/2013/summary/`



ATLAS Collaboration

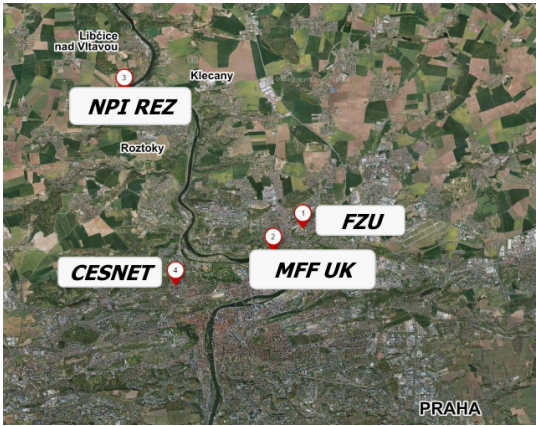
- close to 6000 members
- 176 institutions from 40 countries
- 1500+ publications
  - hundreds of them with acknowledgement to IT4Innovations (already in `https://scs.it4i.cz`)

WLCG



- The mission of the Worldwide LHC Computing Grid (WLCG) is to provide global computing resources for the storage, distribution and analysis of the data generated by the LHC.
  - 1.4 million computer cores
  - 1.5 exabytes of disk
  - 3 exabytes of tape

CZ-Tier-2

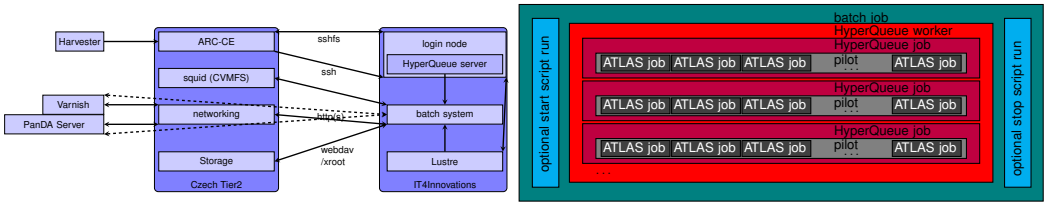


- It is computing center of Institute of Physics of the Czech Academy of Sciences.
- It is distributed over several locations.
- CZ-Tier-2 supports multiple experiments:
  - LHC experiments (ATLAS, ALICE)
  - astroparticle experiments (Pierre Auger Observatory, SST1M)
  - Fermilab experiments (NOvA, DUNE)
  - local users from other experiments
- storage
  - 21PB of grid storage (dCache and xrootd)
  - 1 PB of NFS
- batch
  - compute element: ARC-CE
  - batch system: HTCondor
- networking (all dual-stack, IPv6 preferred)
  - 400 Gbps to LHCONE
  - 100 Gbps to Pasnet/Internet

HyperQueue

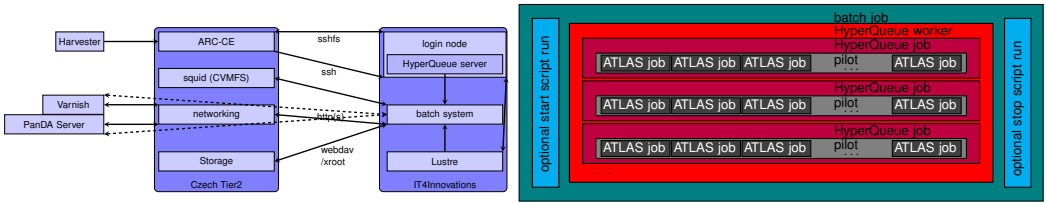
- `https://it4innovations.github.io/hyperqueue/latest/`
- HyperQueue is a tool designed to simplify execution of large workflows on HPC clusters.
- It was developed within IT4Innovations.
- user perspective:
  - It is a small binary with no dependencies which one can just download and run.
  - It works as batch system under my control within a batch system outside of my control.
    * If batch system schedules only whole node jobs (128 cores), I can send four 32 core jobs (for example) to fill that.
    * If batch system enforces re-runnable jobs, I can set them to be not re-runnable.
    * It allows running a mixture jobs with widely ranging resource requirements in the same batch job.
    * ...

Submission system



1. The grid front-end (ARC-CE) shares storage with the Lustre via sshfs connection through a login node and communicates with the batch system via ssh connection (through the same login node).
2. When the ARC-CE receives a job from Harvester (a resource-facing service of workflow management system), it translates the job description into a script that can be run in the batch system, puts necessary files into a folder within sshfs shared area and submits the job via ssh connection to the HyperQueue server running on a login node.
3. The HyperQueue server buffers the jobs and, when there are enough of them to fill a worker node, it submits jobs into the batch system.
4. When the batch job starts, HyperQueue jobs start in it.
5. Software is available through cvmfsexec (`https://github.com/cvmfs/cvmfsexec`).
6. In each HyperQueue job, pilot wrapper starts, launching the pilot.

Submission system



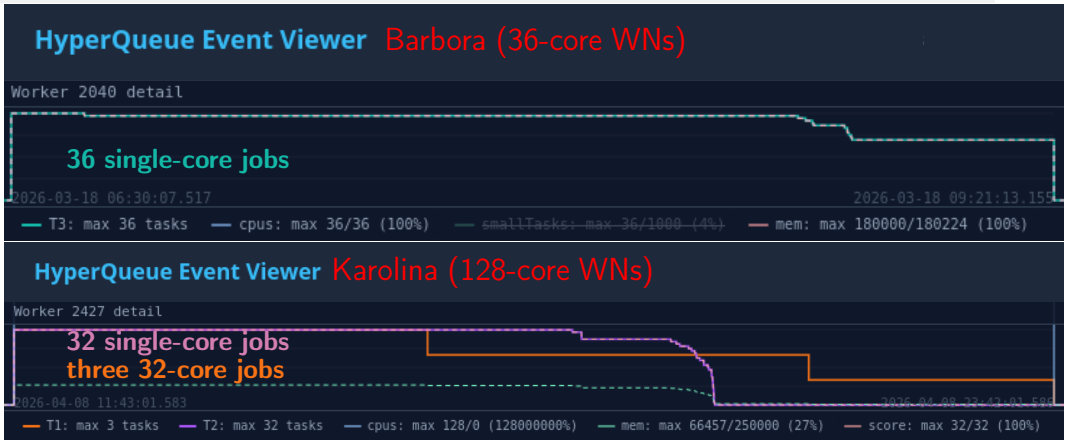
7. Pilot requests payload from PanDA server (which takes care of ATLAS jobs throughout their lifetime).
8. When it receives the payload, it gets input file from the Czech Tier2 storage via xroot or webdav.
9. Then it starts the calculation.
10. If the calculation requires conditions data, they are obtained from external Varnish cache.
11. Contact to external services (PanDA server, Varnish) can be either direct or, if there are limits on available ports, it can hop through the networking of the Czech Tier2.
12. When the payload finishes, it sends outputs to the Czech Tier2 storage via xroot or webdav.
13. When this is finished, pilot will request another payload (if it can expect that the batch queue setting would allow it to finish).

Job mixing

- Until recently, only workloads most suitable for many-core environment were running at IT4Innovations.
- The needs of ATLAS computing evolved and with that came also the need to run more workloads.
- HyperQueue schedulers were updated to allow full utilization of available resources by ATLAS jobs which request different number of CPU cores or memory.
- Filling of batch job can be visualised using the HyperQueue Event Viewer (y-axis is on relative scale, i.e. showing how much of given category is used out of its maximum).
  - on Barbora, we run
    - one 36-core job
    - 36 single-core jobs
  - on Karolina, we run
    - four 32-core jobs
    - three 32-core jobs and 32 single-core jobs



Job mixing


HyperQueue Event Viewer Barbora (36-core WNs)
Worker 2040 detail
36 single-core jobs
2026-03-18 06:30:07.517
2026-03-18 09:21:13.155
T3: max 36 tasks
cpus: max 36/36 (100%)
smallTasks: max 36/1000 (4%)
mem: max 180000/180224 (100%)
HyperQueue Event Viewer Karolina (128-core WNs)
Worker 2427 detail
32 single-core jobs
three 32-core jobs
2026-04-08 11:43:01.583
T1: max 3 tasks
T2: max 32 tasks
cpus: max 128/0 (128000000%)
mem: max 66457/250000 (27%)
score: max 32/32 (100%)

Performance in 2026

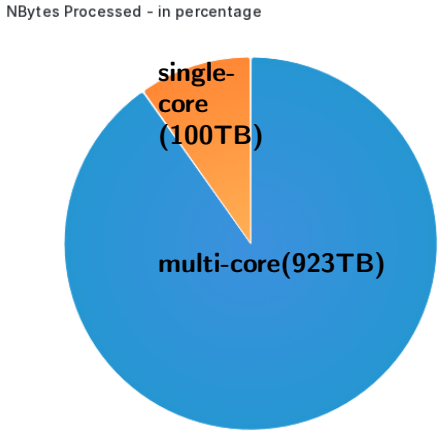
- current project: OPEN-35-22
- We are running on qcpu_preempt batch queue (where jobs are killed when jobs from users with allocation arrive).
  - Allocation is consumed within few days.
  - Our jobs can handle pre-emption well.
    - Jobs are independent (different input, setting, software version, etc.).
    - We have the machinery for automatic submission (ARC-CE and HyperQueue manage job's life cycle).

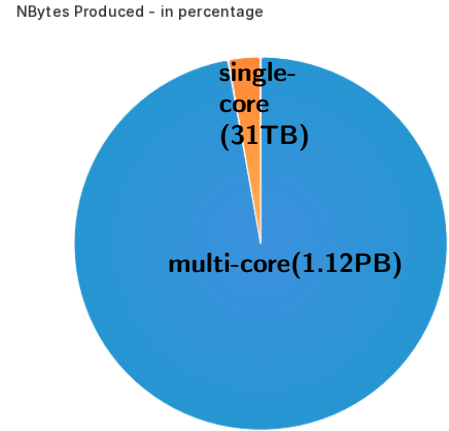
Performance in 2026



- Running on pre-emptive queue means that failure rate is significant.
- HepScore23 (HS23) benchmark was developed for better estimation of processing speed.
  - Combines workloads from multiple particle physics experiment.

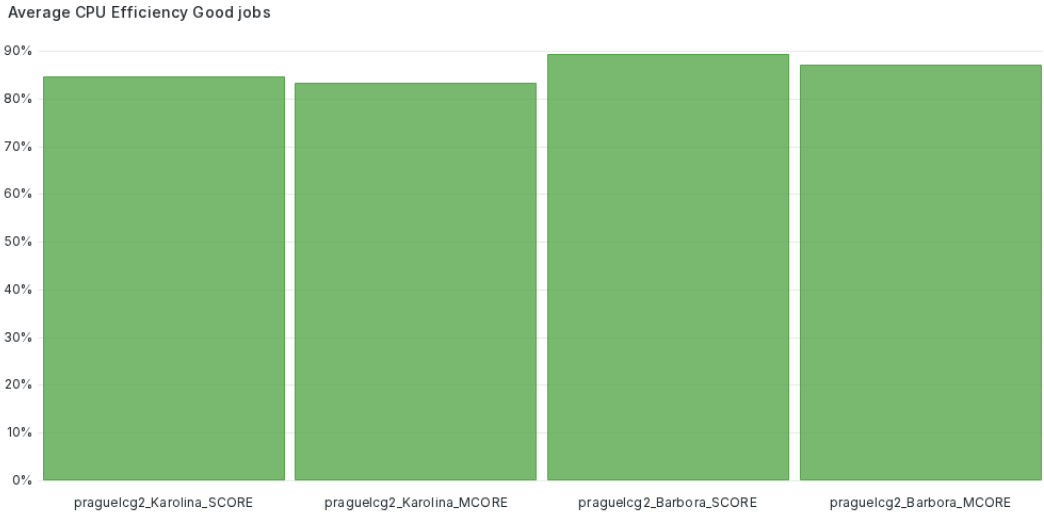
Performance in 2026





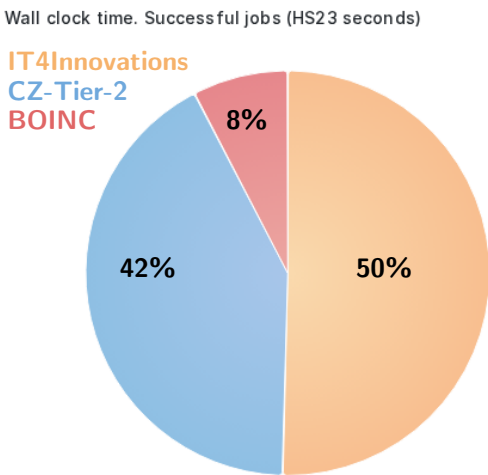
- ∼1 PB of data transferred to IT4Innovations for processing.
- ∼1 PB of output data transferred from IT4Innovations.

Performance in 2026



- CPU efficiency of successful jobs is above 80%, thanks to splitting the batch job into smaller parts by HyperQueue.

Performance in 2026



- comparing all resources contributing to CZ-Tier-2, IT4Innovations provides half

Summary

- HyperQueue allowed us to use finer-grained resource allocation in jobs.
- We can run both single-core and multi-core jobs, i.e. increase the size of available jobs pool.
- It allowed us to use the HPCs to contribute significantly to computing resources delivered to ATLAS.