Updates on usage of the Czech national HPC center

Michal Svatoš[1,*], *Jiří* Chudoba[1,**], and *Petr* Vokáč[2,***]

[1]Institute of Physics of the Czech Academy of Sciences, Na Slovance 1999/2, Prague, 18221, Czech Republic

[2]Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Břehová 7, Prague, 115 19, Czech Republic

Abstract. The distributed computing of the ATLAS experiment at LHC has used computing resources of the Czech national HPC center IT4Innovations for several years. The submission system is based on ARC-CEs installed at the Czech Tier2 site (praguelcg2). Recent improvements of this system will be discussed here. First, there was a migration of the ARC-CE from version 5 to 6 which improves the reliability and scalability. A shared filesystem built on top of sshfs 3.7 no longer represents performance bottleneck. It provided an order of magnitude better transfer performance. New Singularity containers with full software stack can easily fit default resource limits on the IT4I cluster filesystem. A new submission system, allowing sequential running of payloads in one job, was set and adapted to HPC's environment, improving usage on worker nodes with very high number of cores. Overall, the whole infrastructure provides significant contribution to resources provided by praguelcg2.

ATL-SOFT-PROC-2021-006
03 June 2021

1 Introduction

The distributed computing of the ATLAS experiment at LHC [1] has used computing resources of the Czech national HPC center IT4Innovations for several years. In 2020, it was using three HPC systems of the IT4Innovations: Salomon (jobs are being sent there since December 2017), Barbora (used since January 2020), and Anselm (used since February 2020). This provides ATLAS with a significant amount of additional computing resources.

2 Job submission system

The system submitting ATLAS jobs to HPCs of IT4Innovations is shown in Figure 1. The ARC Control Tower (aCT) obtains a job description from the ATLAS workflow management system and submits it to one of the ARC-CE [2] machines installed at the Czech LHC Tier2 site (praguelcg2) [3]. The ARC-CE processes the description and creates a script executable by the PBSpro batch system [4] on the HPC. This script is then submitted into the PBSpro via a ssh connection to an HPC login node. Job auxiliary files are shared between ARC-CE and Lustre storage of an HPC node via sshfs connection. Required job input files are either copied from the local DPM [5] grid storage to the cache located on the Lustre storage of an

[*]e-mail: svatosm@fzu.cz

[**]e-mail: chudoba@fzu.cz

[***]e-mail: petr.vokac@fjfi.cvut.cz

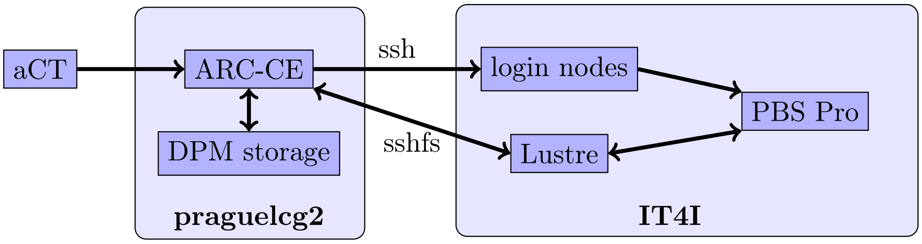


Figure 1. A scheme of the job submission system

HPC using an additional sshfs connection or, if they are already there, symlinked from the cache. Software used by running jobs is also located on the Lustre storage. When the job finishes, the output is transferred from the Lustre storage back to the ARC-CE machine via sshfs and from there to the local DPM grid storage. This submission system is described in more detail in [6].

In 2020, there were five ARC-CE machines submitting jobs to HPCs of IT4Innovations (two to Salomon HPC, two to Barbora HPC, and one to Anselm HPC).

3 Improvements

3.1 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. [7]. The support of version 5 ended. This means no security fixes will be released for ARC 5. As a result, sites were asked to migrate to version 6 [8]. All ARC-CEs were migrated to version 6 in June/July of 2020.

3.2 Sshfs

When this system was set up, it was observed that the connection via sshfs is a bottleneck. While the ARC-CE has 10 Gbps connectivity, a saturation at about 60 Mbps was observed (see Figure 2). The probable cause was that the directory of each job contains many small files. Such situation happens for CentOS7 default version of sshfs (2.10).

In January 2020, sshfs version 3.7 was released [9]. It introduced `max_conns` option which enables the use of multiple connections. Using this version and setting the number of connections to 10 solved the saturation. Figure 3 shows peaks over 1 Gbps with no saturation.

3.3 Containerization

Since 2019, ATLAS can run all its jobs inside of Singularity [10] containers. They expect to have access to the CVMFS [11]. At HPC systems, the CVMFS is not available. What can be done there is to synchronise a part of CVMFS to the shared storage of HPC where jobs can reach it. Necessary software can be also added into a container, forming so-called fat containers [12]. Such containers have all the necessary software and condition data of one release (even though there are still some parts of the CVMFS which need to be synchronised). These containers provide several advantages:

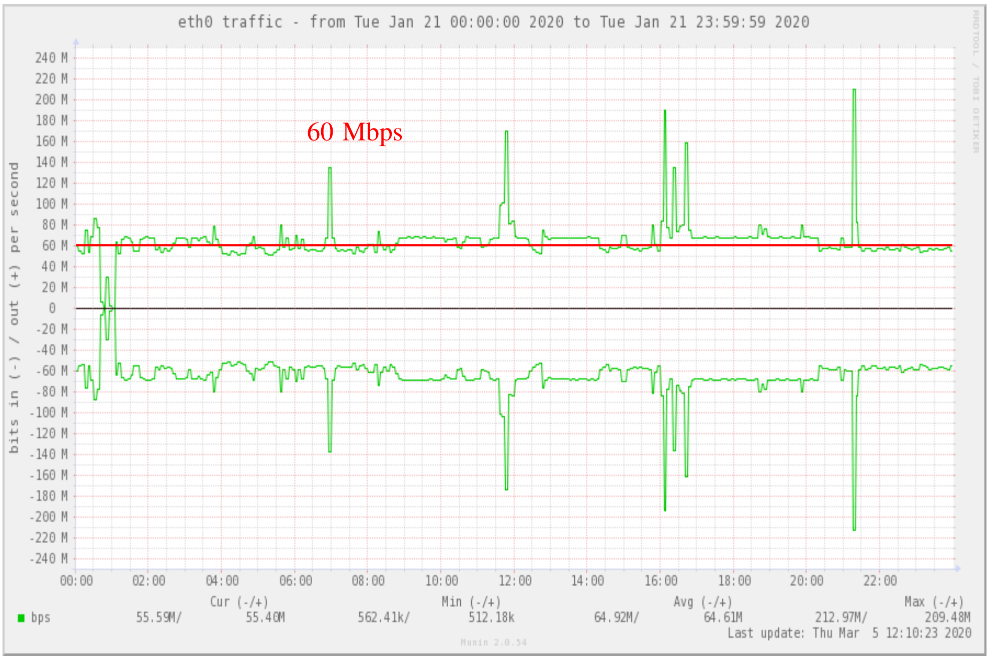


Figure 2. Network performance over 24 hour period with sshfs 2.10. There is network saturation around 60 Mbps.

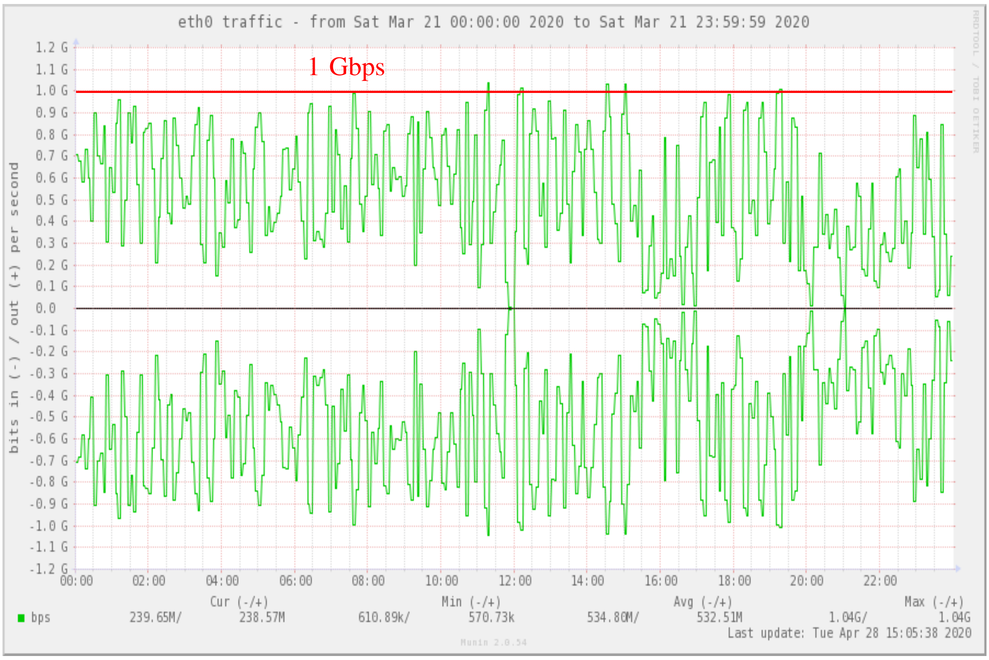


Figure 3. Network performance over 24 hour period with sshfs 3.7 using `-o max_conns=10` option. No saturation is observed.

- Highly specialised expert knowledge is needed to extract one release from the CVMFS. Without it, a bigger part of the CVMFS needs to be synchronised to ensure that jobs have all the necessary software. In practise, this amounts to about 12 million files. Additional complication is HPC's 10 million files limit per user (meaning exceptions need to be requested, approved, and repeatedly renewed). With a fat container, about 10 million files were removed from each HPC, making exceptions unnecessary. Only the software necessary for containers usage remained.
- There were also occasional failures caused by timeouts to Squid at praguelcg2, which were hard to reproduce and debug. The fat containers do not need to contact a Squid server as they have all the necessary condition data packed inside.

3.4 Long jobs

With increasing number of CPU cores (the new HPC of IT4Innovations is supposed to have worker nodes with 128 cores), the duration of workload processing shrinks. To improve the opportunistic usage of an HPC, the length of running jobs needs to increase. There are several ways how to do it.

3.4.1 Dedicated tasks

The simplest way would be to have dedicated tasks with more events per job. But this costs time of people who manage tasks. Over the HPC's lifespan, it can represent significant amount of expert work.

3.4.2 Parallel jobs

Next option would be to have several jobs to run in parallel. This can be done using local installation of Harvester [13] (replacing a compute element) and its Many-to-1 workflow [14]. While the Harvester is able to replace a compute element in terms of job submission, it does not provide its other functionalities like reporting the accounting data into the APEL [15].

3.4.3 Sequential jobs

The last option would be to have several jobs running sequentially. This way, when a job is started in the batch system, it requests a payload. When the payload finishes, it requests another one as long as it is within allowed time limit. This can be done if worker nodes have outbound connectivity. Such system works on many grid sites as those can open appropriate ports. The problem is that an HPC is a rather closed environment. Even though HPCs of IT4Innovations have outbound connectivity, it is basically only through http(s) ports. That means ATLAS jobs cannot contact PanDA (Production and Distributed Analysis system) [16] servers to get payload or do stage-in/out. Thus, several modifications and workarounds are needed to make the system work also on HPC.

Currently, the system works as follows (see Figure 4). The ARC-CE receives a pilot job, processes it, 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 does stage-in of input file(s) from DPM on praguelcg2 via webdav. This is done inside of the modified Rucio [17] container. The container is modified in such a way that it always prefers usage of webdav (i.e. http, which uses an allowed port) irrespectively of the setting of protocol priority for praguelcg2 storage

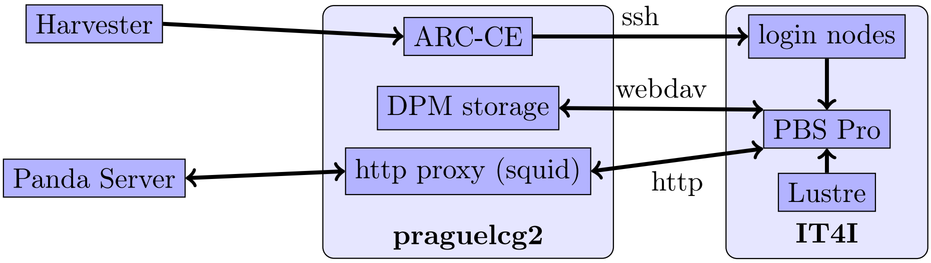


Figure 4. A scheme of the job submission system

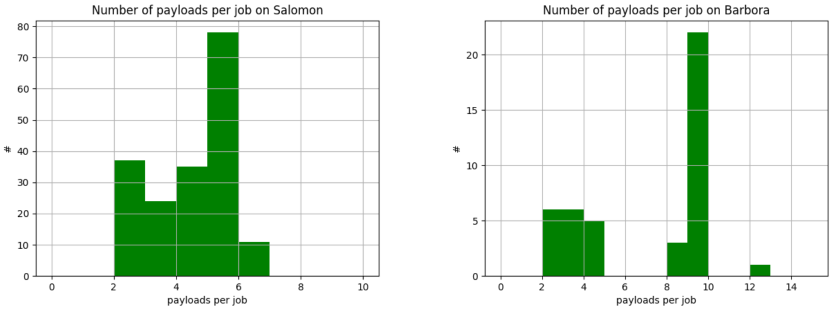


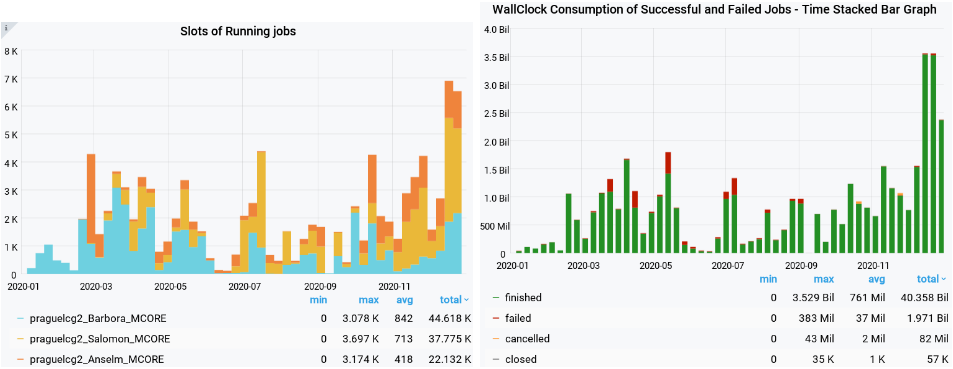
Figure 5. Number of payloads in jobs on currently available HPCs. Plots ignore 0 payloads (empty pilot) and 1 payload (failing stage-in/stage-out during testing). Salomon HPC can usually run up to five payloads and Barbora HPC nine.

(which does not always use webdav as primary protocol). When the payload finishes, it does stage-out of outputs to DPM on praguelcg2 via webdav. Again, this is done inside of the modified Rucio container. Pilot will then request another payload (if it can expect that the payload finishes within a batch queue allowed walltime).

This system ran for a couple of weeks at low rate. There are several observations that can be made.

- Analysis of timing in logs of several jobs gives promising results. A job spends few seconds to few minutes on file transfers (stage-in, stage-out), about two minutes on single-core part of the simulation and about two minutes on merging of outputs. These numbers should be similar for all simulation jobs. The multi-core part of the simulation can vary significantly with average time of two hours on 24 core machines. The time with fully utilized 128 cores on new HPC machine is going to be shorter but still the longest part of whole calculation. That will still provide a decent overall CPU utilization efficiency.
- Number of payloads per job depends mostly on maximal allowed length of a job (which is same for all HPCs of IT4Innovations) and CPU of the worker node. Figure 5 shows that the Salomon HPC is usually able to run up to five payloads and Barbora HPC nine.

This system still has problems which need to be investigated. For example, there are some proxy length issue. Also, at certain times, there are lots of empty pilots (i.e. pilots not getting any payload).



	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

	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

Figure 6. ATLAS usage of the IT4Innovations machines. On the left, there are slots of running jobs. As those are opportunistic resources, they fluctuate significantly. On the right, there is combined finished and failed walltime (in seconds multiplied by number of cores). The failure rate is actually rather low.

4 Performance

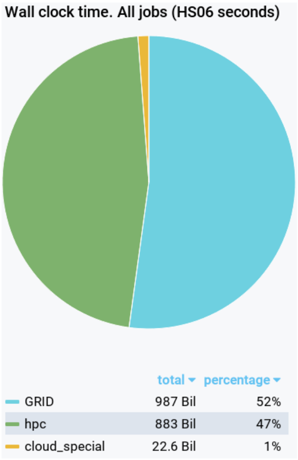
In 2020, ATLAS was using three HPCs of IT4Innovations. The left plot on Figure 6 shows peaks in number of running job slots characteristic for opportunistic nature of the HPC usage. The right plot shows used wallclock (total time jobs occupied computing resources).

The failure rate is rather low and comes as short peaks often coming from infrastructure or central problems. Figure 7 shows wallclock in HS06 provided by various resource types of the praguelcg2 in 2020. The opportunistic resources provided almost half of the wallclock provided by the site.

5 Summary and Conclusion

The system submitting jobs to the the Czech national HPC center IT4Innovations received several upgrades. The ARC-CE machines submitting jobs were migrated to a new supported version. With new version of sshfs, there is no saturation of the network connection. Usage of fat containers leads to decrease in number of files on the HPC by 10 million and prevents Squid connection timeouts. A system submitting payloads sequentially has been adapted to the closed environment of HPCs and successfully submits jobs. It also gives promising result for running on worker nodes with very high number of cores.

The HPCs of the IT4Innovations contributed significantly to the distributed computing of the ATLAS experiment at LHC. Specifically, they contributed almost half of walltime (in HS06) provided by the Czech computing resources in 2020. [a]



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

Figure 7. Wallclock usage (in HS06) of praguelcg2 resources in 2020. The HPC provides almost half of contributed walltime.

[a] 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.
We gratefully acknowledge National Supercomputing Center IT4Innovations (IT4I) for computing time provided by their supercomputers.

References

[1] ATLAS Collaboration, JINST **3**, S08003 (2008)

[2] D. Cameron, A. Filipčič, W. Guan, V. Tsulaia, R. Walker, T. Wenaus (ATLAS), J. Phys. Conf. Ser. **898**, 052010 (2017)

[3] J. Chudoba, M. Adam, D. Adamová, T. Kouba, A. Mikula, V. Říkal, J. Švec, J. Uhlířová, P. Vokáč, M. Svatoš, J. Phys. Conf. Ser. **898**, 082034 (2017)

[4] *PBS professional open source project*, https://www.pbspro.org

[5] A. Manzi, F. Furano, O. Keeble, G. Bitzes, J. Phys. Conf. Ser. **898**, 062011 (2017)

[6] M. Svatos, J. Chudoba, P. Vokac, EPJ Web Conf. **214**, 03005 (2019)

[7] *Main changes in ARC 6 compared to ARC 5*, http://www.nordugrid.org/arc/arc6/common/changelog/arc5_to_arc6_changes.html, [accessed 2020-11-1]

[8] *ARC middleware 5 end of support, migration to ARC 6*, https://wiki.egi.eu/wiki/Agenda-2020-05-11#ARC_Middleware_5_end_of_support.2C_migration_to_ARC_6, [accessed 2020-11-09]

[9] *sshfs changelog*, https://github.com/libfuse/sshfs/blob/master/ChangeLog.rst, [accessed 2020-11-11]

[10] G.M. Kurtzer, V. Sochat, M.W. Bauer, PLoS ONE **12** (2017)

[11] A. De Salvo, A. De Silva, D. Benjamin, J. Blomer, P. Buncic, A. Harutyunyan, A. Undrus, Y. Yao (ATLAS), J. Phys. Conf. Ser. **396**, 032030 (2012)

[12] N. Ozturk, A. Undrus, M. Vogel, A. Forti, Tech. rep., CERN, Geneva (2021), paper was submitted as ATLAS-COM-CONF-2021-002 on 1/2/2021, now submitted here under ATLAS-COM-SOFT., https://cds.cern.ch/record/2750998

[13] *Harvester*, https://github.com/HSF/harvester/, [accessed 2021-02-04]

[14] *Harvester workflows*, https://github.com/HSF/harvester/wiki/Workflows, [accessed 2020-11-24]

[15] *EGI production accounting*, https://accounting.egi.eu/, [accessed 2021-02-04]

[16] F.H. Barreiro Megino, K. De, A. Klimentov, T. Maeno, P. Nilsson, D. Oleynik, S. Padolski, S. Panitkin, T. Wenaus (ATLAS), J. Phys. Conf. Ser. **898**, 052002 (2017)

[17] M. Barisits, T. Beermann, F. Berghaus, B. Bockelman, J. Bogado, D. Cameron, D. Christidis, D. Ciangottini, G. Dimitrov, M. Elsing et al., Computing and Software for Big Science **3**, 11 (2019)