

# ATLAS utilisation of the Czech national HPC center

Michal Svatos<sup>1,\*</sup>, Jiri Chudoba<sup>1,\*\*</sup>, and Petr Vokac<sup>2,\*\*\*</sup>, on behalf of the ATLAS Collaboration

<sup>1</sup>Institute of Physics of the CAS, Na Slovance 1999/2, Prague, 18221, Czech Republic

<sup>2</sup>Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Brehova 7, Prague, 115 19, Czech Republic

**Abstract.** The Czech national HPC center IT4Innovations located in Ostrava provides two HPC systems, Anselm and Salomon. The Salomon HPC is amongst the hundred most powerful supercomputers on Earth since its commissioning in 2015. Both clusters were tested for usage by the ATLAS experiment for running simulation jobs. Several thousand core hours were allocated to the project for tests, but the main aim is to use free resources waiting for large parallel jobs of other users.

Multiple strategies for ATLAS job execution were tested on the Salomon and Anselm HPCs. The solution described herein is based on the ATLAS experience with other HPC sites. ARC Compute Element (ARC-CE) installed at the grid site in Prague is used for job submission to Salomon. The ATLAS production system submits jobs to the ARC-CE via ARC Control Tower (aCT). The ARC-CE processes job requirements from aCT and creates a script for a batch system which is then executed via ssh. Sshfs is used to share scripts and input files between the site and the HPC cluster. The software used to run jobs is rsynced from the site's CVMFS installation to the HPC's scratch space every day to ensure availability of recent software.

## 1 Introduction

The Czech National Supercomputer Center IT4Innovations in Ostrava operates the Salomon HPC system which is the most powerful computer in the Czech Republic; it is listed in Top500 and currently ranked 87th in the world. Salomon was built in 2015, providing 2 PFLOPs in peak performance. It consists of 1008 computational nodes with 24 cores of Intel Xeon E5 CPUs and 128 GB of RAM per node interconnected with Infiniband (56 Gbps). In addition, 432 nodes contain 61 core Intel Xeon Phi accelerators. In total, there are 24192 cores of computational nodes with an additional 52704 accelerator cores. Salomon nodes run on CentOS 6 with PBSpro as a batch system. Nodes have very limited external connectivity through http/https proxy.

The ATLAS Experiment in CERN [1] uses the Salomon cluster in opportunistic fashion via the Czech Tier2 site (pragueclg2) [2]. Only non-accelerated nodes are available for opportunistic usage and unlike other opportunistic HPC resources, there is no job pre-emption. The

---

\*e-mail: Michal.Svatos@cern.ch

\*\*e-mail: Jiri.Chudoba@cern.ch

\*\*\*e-mail: petr.vokac@cern.ch



HPC batch scheduler selects jobs of projects without dedicated computing time allocation to increase supercomputer utilization efficiency by filling empty batch slots with opportunistic jobs. Jobs waiting for resources for a long time are killed in the batch system and reassigned by ATLAS production system to another site to ensure timely completion of tasks.

The first successful ATLAS job submitted to the Salomon via ARC-CE finished in December 2017 and since then Salomon continuously and significantly contributes to Czech Tier2 ATLAS production output.

## 2 ARC-CE

Traditional pilot based grid jobs do not always deal well with the restrictive HPC environment. ATLAS developed ARC Control Tower (aCT) to be more flexible with utilizing supercomputers with specific configurations. ATLAS jobs coming from aCT are transformed by the ARC-CE [3] computing element into a script suitable for HPC's PBSPro batch system. The environment required by jobs is set via RunTime Environments (RTE) (see section 3). The computing element also downloads input files which are stored in the session directory together with the startup scripts and subsequently uploads output files. The standard ARC-CE interface for the Torque batch system was modified to accommodate differences with PBSpro batch system on Salomon and job management that must be done with ssh through an HPC login node. Another ARC-CE modification was necessary to provide HTTP proxy configuration for access to condition data:

```
echo "export 'http_proxy=http://prague1cg2SquidIP:443'" »
$LRMS_JOB_SCRIPT
echo "export 'all_proxy=http://prague1cg2SquidIP:443'" »
$LRMS_JOB_SCRIPT
```

The changes in Torque commands (to go through ssh) are as follows:

### qsub

```
/usr/bin/ssh -i path_to_id_rsa user@loginnode.salomon.it4i.cz
"/opt/pbs/default/bin/qsub -A DD-13-2 $@"
```

### qstat

```
/usr/bin/ssh -i path_to_id_rsa user@loginnode.salomon.it4i.cz qstat $@
```

### qdel

```
/usr/bin/ssh -i path_to_id_rsa user@loginnode.salomon.it4i.cz qdel $@
```

### qmgr

```
/usr/bin/ssh -i path_to_id_rsa user@loginnode.salomon.it4i.cz qmgr
\"$@\
```

### pbsnodes

```
/usr/bin/ssh -i path_to_id_rsa user@loginnode.salomon.it4i.cz pbsnodes
$@
```

### 3 RunTime Environment

RunTime Environments (RTEs) are a way to accommodate environments which are specific for given site/experiment/job/etc. RTE itself is a shell environment initialization script. As it is necessary for a job to run, its location must be directly accessible from a WN (see section 4). The following RTEs are required by ATLAS jobs:

#### ENV/PROXY

```
case "$1" in
0) cat $joboption_controlldir/job.$joboption_gridid.proxy >
$joboption_directory/user.proxy
;;
1) export X509_USER_PROXY=$RUNTIME_JOB_DIR/user.proxy
;;
esac
```

#### APPS/HEP/ATLAS-MULTICORE-1.0

```
case "$1" in
0 )
    if [ -z "$joboption_nodeproperty_0" ]; then
        export joboption_nodeproperty_0=" -N 1"
    else
        export joboption_nodeproperty_0="$joboption_nodeproperty_0 -N 1"
    fi
esac
export RUNTIME_ENABLE_MULTICORE_SCRATCH="yes"
```

#### APPS/HEP/ATLAS-SITE

```
export VO_ATLAS_SW_DIR=/cvmfs/atlas.cern.ch/repo/sw
export FRONTIER_SERVER="(serverurl=http://atlasfrontier-ai.cern.ch:8000/atlr)
(serverurl=http://atlasfrontier2-ai.cern.ch:8000/atlr)
(serverurl=http://atlasfrontier1-ai.cern.ch:8000/atlr)
(serverurl=http://ccfrontier.in2p3.fr:23128/ccin2p3-AtlasFrontier)
(serverurl=http://ccfrontier01.in2p3.fr:23128/ccin2p3-AtlasFrontier)
(serverurl=http://ccfrontier05.in2p3.fr:23128/ccin2p3-AtlasFrontier)
(proxyurl=http://prague1cg2SquidIP:443)"
```

### 4 Shared folder

Session, runtime, and cache directories are mounted via sshfs on dedicated scratch space on Salomon's Lustre filesystem. Syntax of the sshfs command is following:

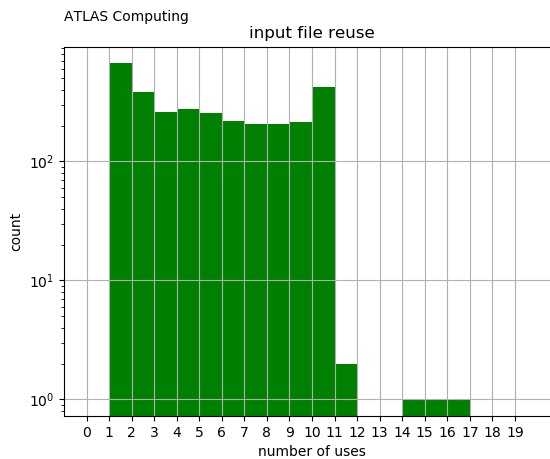
```
sshfs user@loginnode.salomon.it4i.cz:path_on_ARC-CE path_on_HPC
-o reconnect -o allow_other -o workaround=rename -o idmap=file -o
uidfile=path_to_sshfs-hpc.uidmap -o gidfile=path_to_sshfs-hpc.gidmap
-o nomap=ignore -o sftp_server=path_to_sftp-server -o
ServerAliveInterval=30 -o ServerAliveCountMax=2 -o
IdentityFile=path_to_id_rsa -s -o nonempty
```

## 5 Software

The CVMFS [4] is available on the ARC-CE and required files are synchronized with the scratch area of the HPC via `rsync` once a day because CVMFS is not available on worker nodes. Standard ATLAS jobs use executables, libraries, and data from the `atlas.cern.ch` repo (`sw/local`, `sw/software/21.0`, and `ATLASLocalRootBase` directories) and from the `sft.cern.ch` repo (`binutils` directory) and that leads to approximately 0.5 TB and 10 million files that must become available through HPC's Lustre shared filesystem.

## 6 Cache

The analysis of input files reuse shows that many input files are used more than once. The analysis was performed on jobs that finished successfully between end of December 2017 and the beginning of March 2018. During this time period, there were 15k successful jobs (which means 15k input files) but that represents only 3k unique files (Figure 1). Some of these files were used up to 17 times. This was sufficient motivation to deploy an ARC-CE cache located on the shared file system.



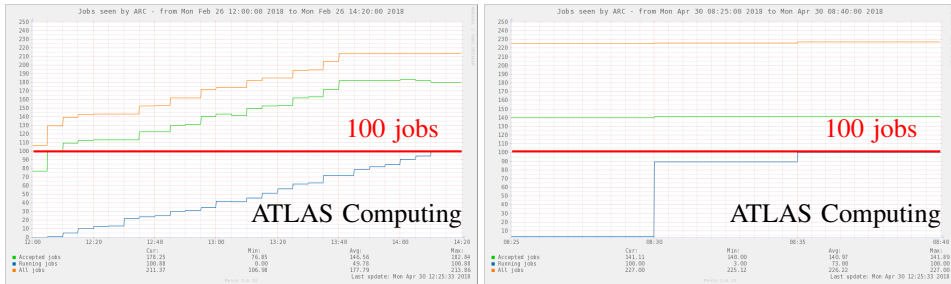
**Figure 1.** Cache reuse factor. Each bin represents number of files which were used  $N$  times where  $N$  is the number on the x axis.

To introduce caching, modification of `sshfs` sharing was necessary as the ARC-CE needs to be able to make both soft and hard links within the cache area. `OpenSSH` was downloaded and compiled on the HPC. Then parameter `-o sftp_server=path_to_sftp_server` was added to the `sshfs` command.

The reduction of job start-up time is significant. Before the cache was deployed, it took  $O(1)$  h (left plot on Figure 2) to create jobs for all 100 slots. With the cache, it takes  $O(1)$  min (right plot on Figure 2). Each bin on plots on Figure 2 is 5 min. Cache size is  $O(1)$  TB because only files which were not used in the last 60 days are deleted.

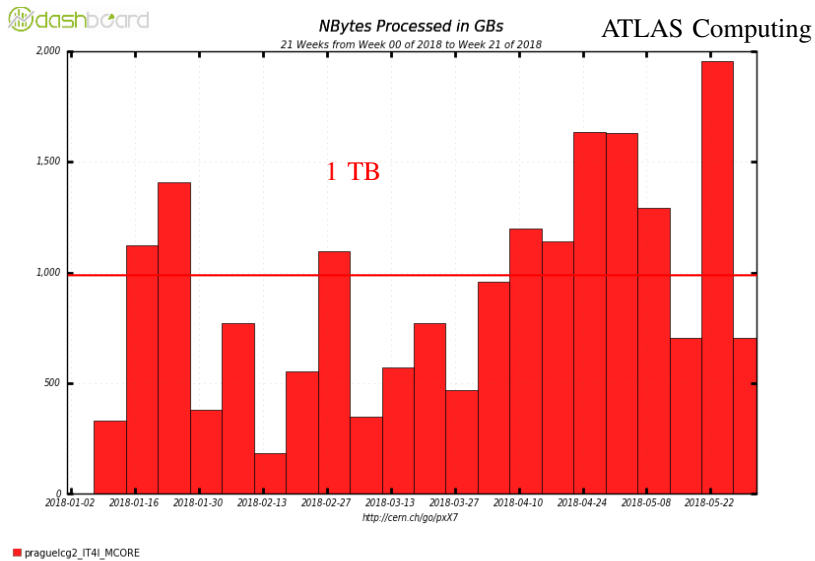
## 7 Job statistics (January-May 2018)

The HPC is running only one ATLAS workload, simulation jobs. They have the best ratio of CPU intensity to I/O, i.e. they have relatively small input and output while having high



**Figure 2.** Time to fill 100 slots without cache (left) and with cache (right). Each bin is 5 min. The blue line represents number of running jobs.

demand on CPU. That is why during a week, the amount of processed input (Figure 3) and produced output (Figure 4) is usually below or around 1 TB while processing several millions events (Figure 5).



**Figure 3.** Amount of data processed every week

The plot of finished and failed jobs (Figure 6) shows increased failure rate until first half of February and then again in April:

- The former was caused by PBS file handle leak, i.e. in some cases which seemed quite random from batch system user point of view, PBS opened resourcedef file (internal PBS file) and handed it over to job. When job checked which files were open, it found it and failed with:

```
boost::filesystem::status: Permission denied:
"/var/spool/PBS/mom_priv/hooks/resourcedef" (C++ exception of type
boost::filesystem::filesystem_error)
```

As a workaround, the following function which closes the file handles was added into

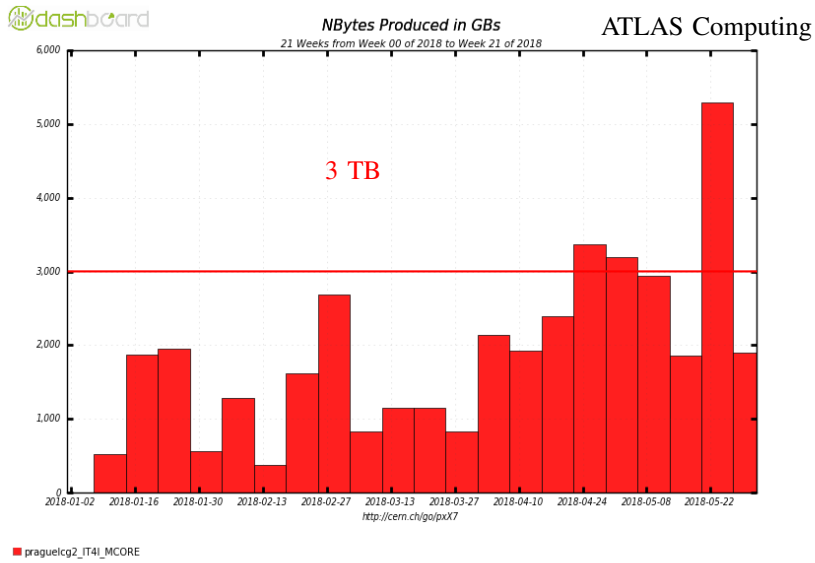


Figure 4. Amount of data produced every week

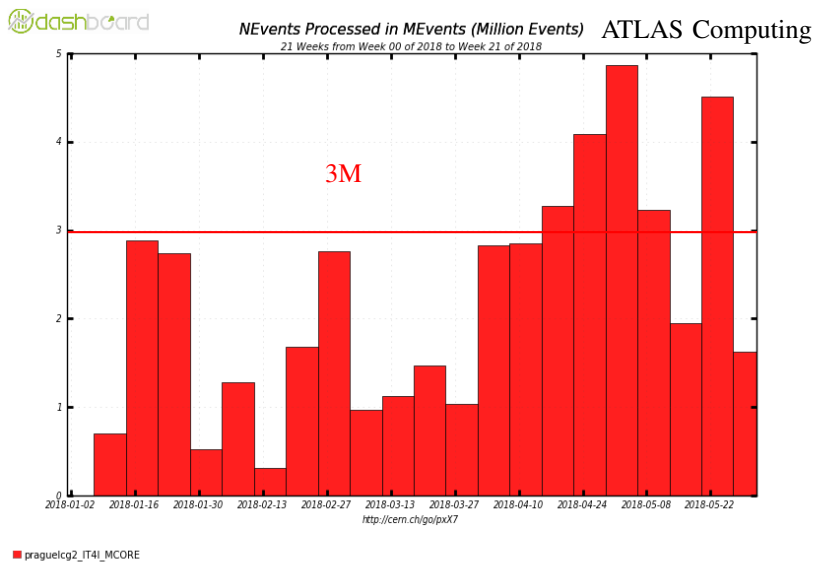


Figure 5. Number of events processed every week

```

submission script:
for f in /proc/self/fd/*
do
  l=$(readlink -m $f)
  if [ "$l" = /var/spool/PBS/mom_priv/hooks/resourcedef ] ; then
    eval "exec $(basename $f)>&-"
  fi
done

```

- The latter failures were caused by a software installation issue. But as the wallclock plot (Figure 7) shows, it did not waste significant amount of resources.

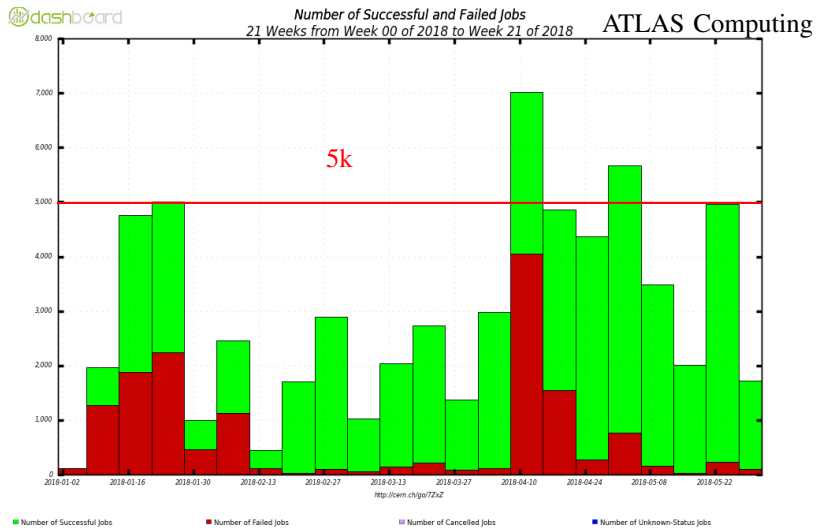


Figure 6. Number of finished and failed job every week

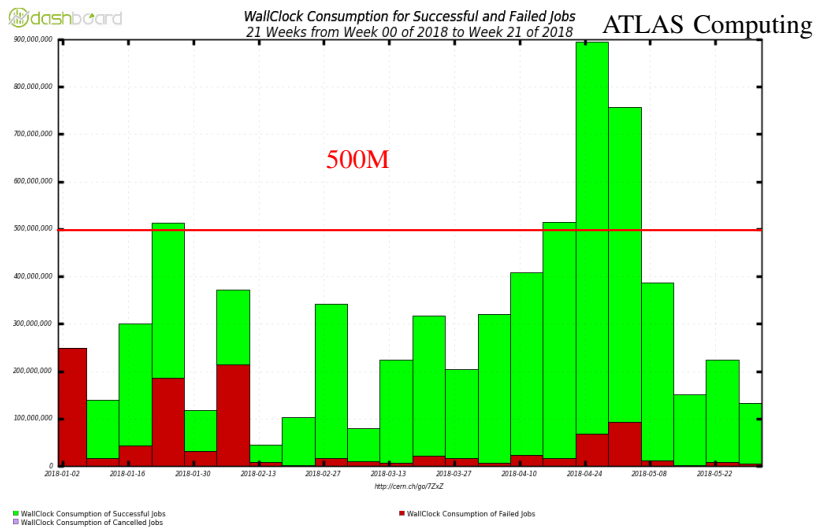


Figure 7. Wallclock consumption of finished and failed job every week

The plot with CPU consumption of all good jobs in different active prague1cg2 queues (Figure 8) shows that the HPC provides significant amount of resources (about 15%).

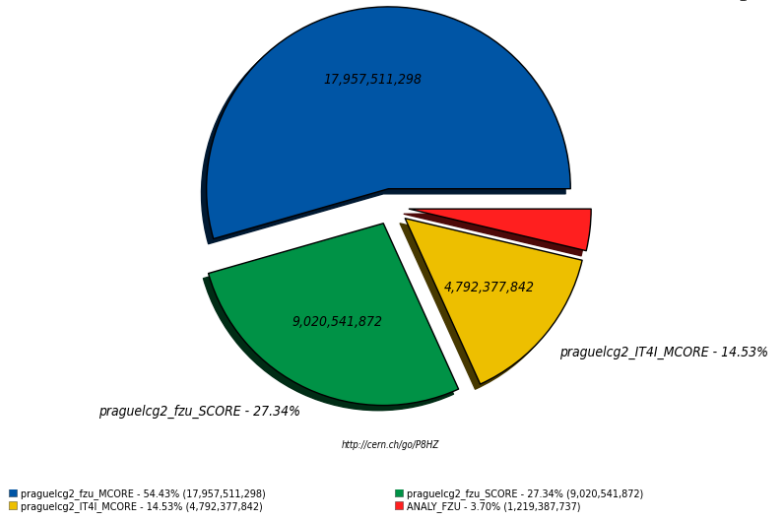


Figure 8. CPU consumption of good jobs for all active prague1cg2 queues

## 8 Acknowledgement

Computing resources as 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.

## 9 Conclusion

The ARC-CE at prague1cg2 was successfully configured and customized for job submission into Salomon HPC to exploit its opportunistic capacity. During the time period between January and May 2018, it provided significant additional computing capacity to Czech Tier2 site.

## References

- [1] ATLAS Collaboration, JINST **3**, S08003 (2008)
- [2] 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)
- [3] M. Ellert, M. Grønager, A. Konstantinov, B. Kónya, J. Lindemann, I. Livenson, J.L. Nielsen, M. Niinimäki, O. Smirnova, A. Wäänänen, Future Gener. Comput. Syst. **23**, 219 (2007)
- [4] J. Blomer, C. Aguado Sanchez, P. Buncic, A. Harutyunyan, J. Phys. Conf. Ser. **331**, 042003 (2011)