

ATLAS Installations on CVMFS

CernVM Workshop 2022

12 -13 Sep 2022

Nikhef

Oana Vickey Boeriu

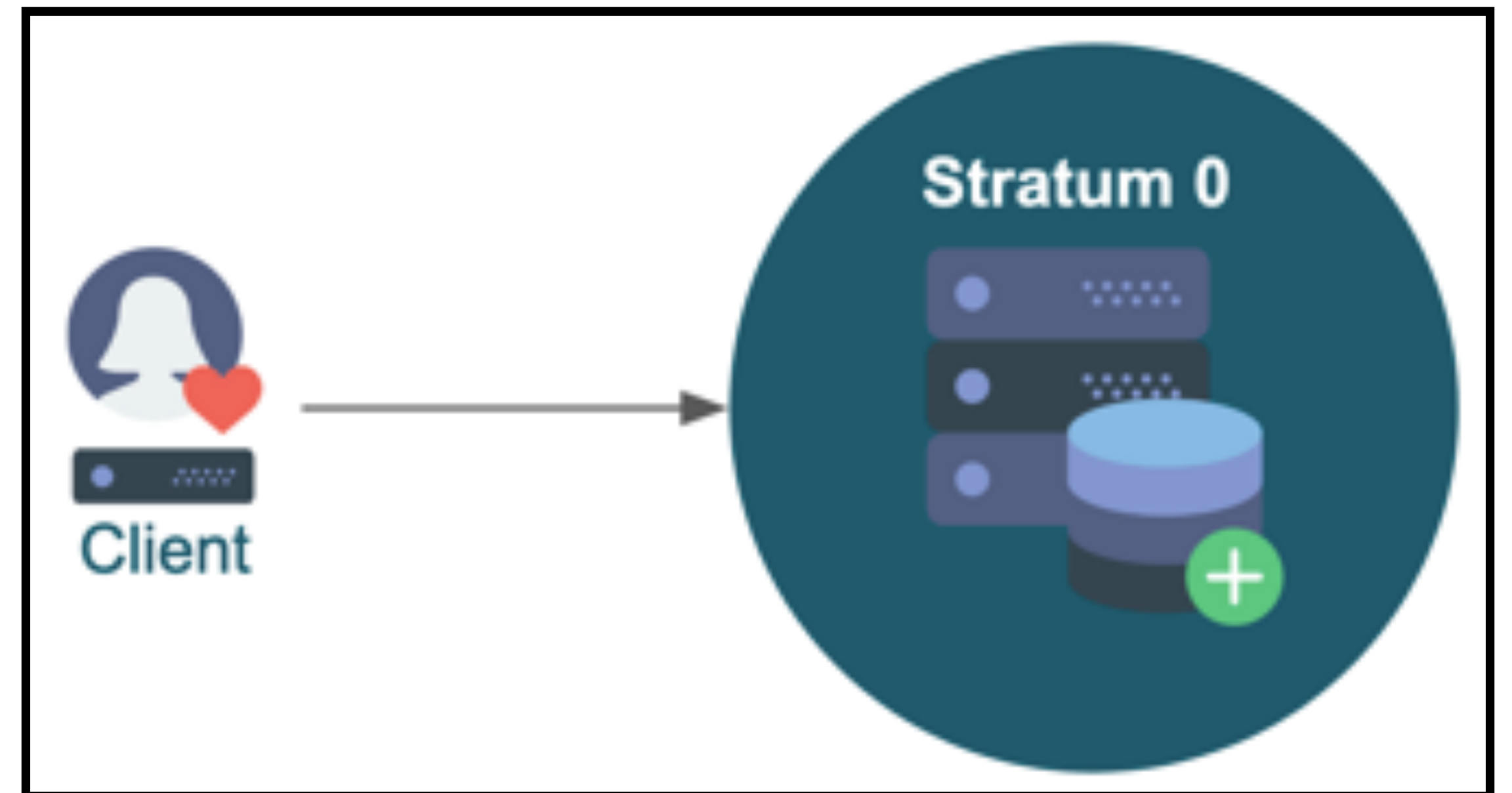
on behalf of the atlas-comp-ASCI/G



The
University
Of
Sheffield.

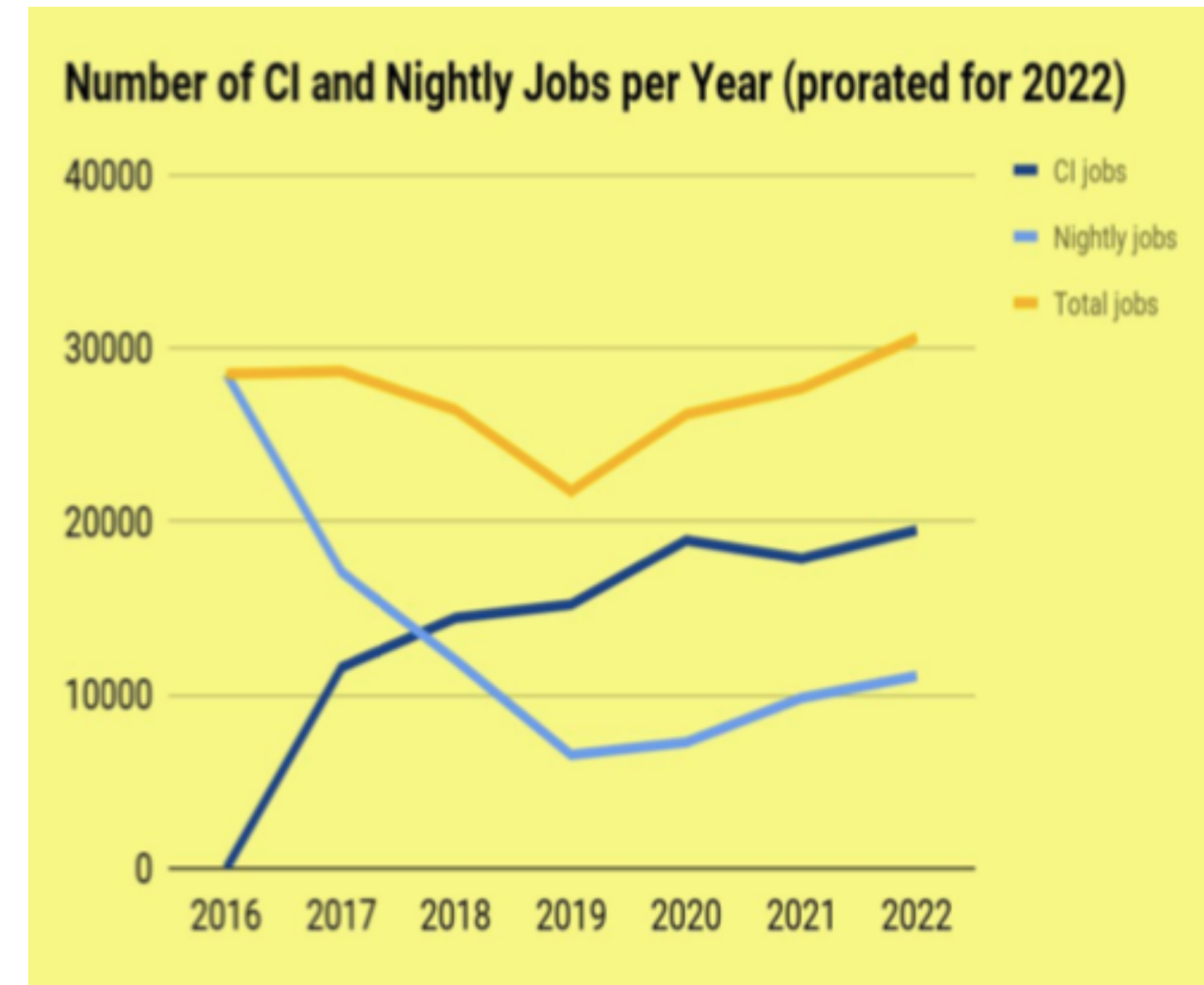
Overview

- Nightlies Overview & ATLAS Build Farm
- Installation Infrastructure & Overview
- Installations Status
- Installations & CVMFS Actions
 - Installation Times
- Parallel Installations & Actions
 - Installation Times
 - Migration
 - Future Installation Software Improvements
- Desired CVMFS Improvements



Nightlies Overview

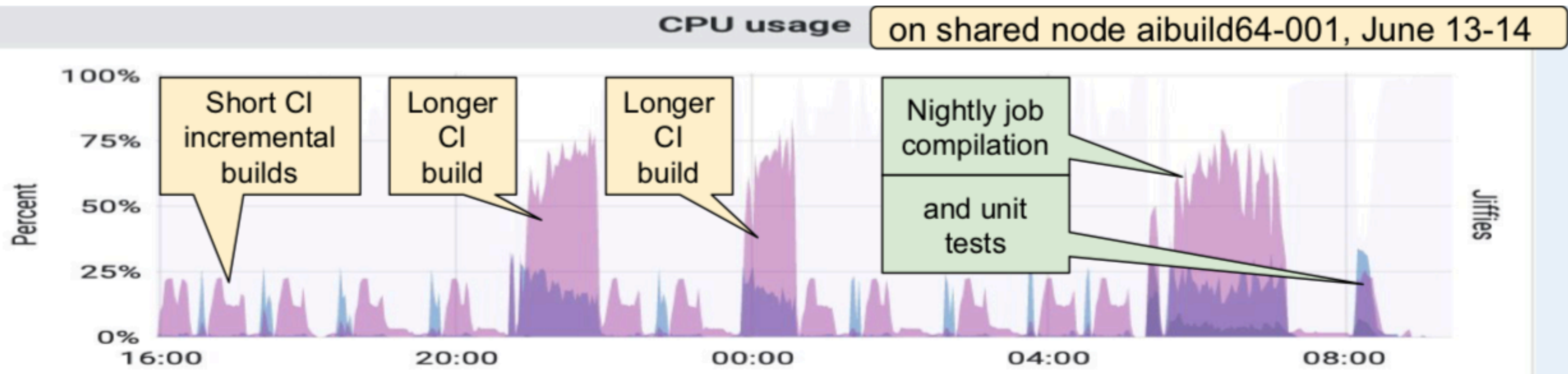
- Stability: ~99% jobs completed without tech problems
 - Ten new 64-core machines added to the build farm
 - Reduced Athena nightly job times from 10 to 3.8 hours
 - Build jobs scheduled through Jenkins
- Shared CI/Nightlies mechanism used
- Full support on the aarch64 platform
- Investigative builds on CentOS9 ongoing
- Improved BigPanDA-based monitoring



37 nightly branches are supported - a 20% increase since 2022. New builds probe new compilers, optimization options, new versions of externals, as well as the new 22.0 git branch

Build Farm: efficient use

- Our build nodes are shared between the CI and the Nightly systems
 - through available Jenkins on-demand launching functionality
- Sharing mechanism works already on the OpenLab ARM machine for SPI and ATLAS jobs (proposed by the SPI team)



Installation Infrastructure

- Repo name: atlas-nightlies.cern.ch - hosting offline sw production crucial for the dev team
- CC7 publisher - lxcvmfs138 since Oct 2020; S3 10TB storage space; 16 CPUs, 28 GB (mem), 160 GB disk size

```
Account: v1
Containers: 1
Objects: 6783982
Bytes: 3722275621414
```

- nightlies used on the grid for different tests and also to produce MC test samples if necessary
- ROOT files reside on the same repo, again used by different tests
- Our installation software code uses ayum (wrapper around yum):
 - downloads, configures & installs python 2.7
 - downloads & installs a virtual environment
 - creates virtual env with all required 3rd party modules

Nightlies Installation Overview

- The installation is triggered from a Jenkins nightly build job



- Three types of installations:
 - opt (5GB), dbg (50GB) and opt-dbg (44GB)
- Installed under:
 - /cvmfs/atlas-nightlies.cern.ch/repo/sw/<branch_project_platform>/<timestamp>/

Installations Status

- Our CVMFS installs are stable
- Failures seen ~only for missing dependencies or corrupt RPM database
 - Missing dependencies - either unreachable remote repositories (“time out”) or actually missing RPMs
 - Corrupt RPM DB requires manual removal, but is quite rare (~1-2 per year)
- Since moving to the CC7 server, update of Stratum 1 failed on one occasion
- Monitoring of Stratum 1 replication is available [here](#)

ATLAS Nightly Builds

[Browse this repository \(experimental\)](#)

Repository atlas-nightlies.cern.ch	
Revision	54005
Oldest stratum1 revision	54005
Last modified	12th September 2022 2:22:31 am
Whitelist expiry date	2022-11-08T09:00:00.000Z

Stratum 1

<div>CERN</div> <div>✓</div> <div>Revision: 54005</div> <div>Last Modified: 12th September 2022 2:22:31 am</div> <div>http://cvmfs-stratum-one.cern.ch/cvmfs</div>	<div>SDCC by RACF at BNL</div> <div>✓</div> <div>Revision: 54005</div> <div>Last Modified: 12th September 2022 2:22:31 am</div> <div>http://cvmfs.sdcc.bnl.gov/cvmfs</div>
<div>KEK</div> <div>✓</div> <div>Revision: 54005</div> <div>Last Modified: 12th September 2022 2:22:31 am</div> <div>http://cvmfs-stratum-one.cc.kek.jp/cvmfs</div>	

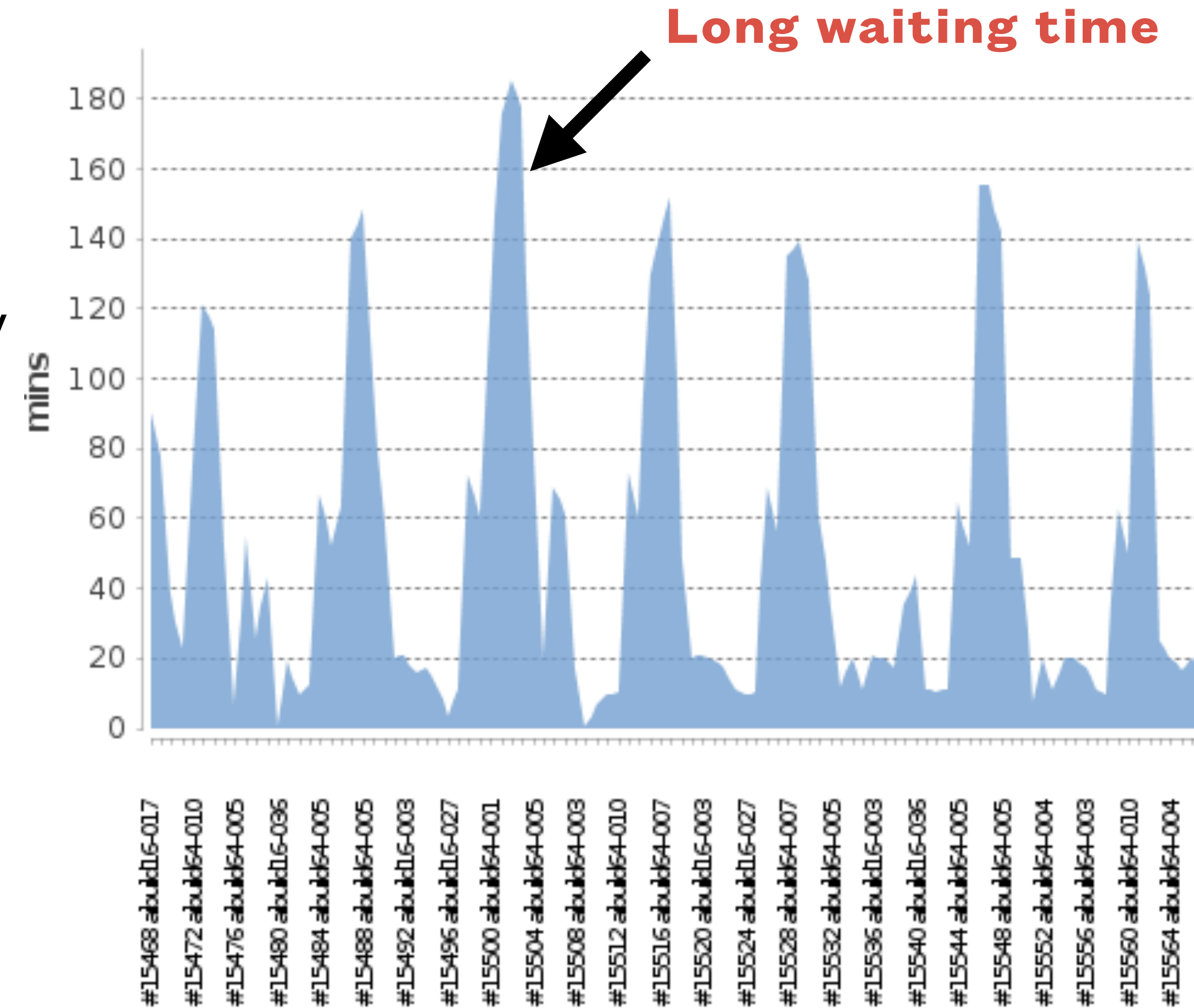
Installations & CVMFS Actions

- The following actions need a transaction + publication at random times every day/night:
 - The nightly builds produce RPM files which are automatically installed
 - ROOT files updates/deletions/additions for tests performed every two hours
 - One AMI tags file - updated after each RPM installation
 - Removal of installations older than 30 days - performed via a cronjob once a day
 - Installing privately build RPMs (outside of the regular nightly system) - when necessary
 - Keep installations longer than 30 days (adding a .keep file under a particular installation) - when necessary
 - Development LCG nightly builds are installed on cvmfs daily as well
 - Performing the garbage collection - daily via a cronjob



Installation Times

- Having ca 40 nightly builds that need to be installed on CVMFS and only one manager creates long queues
- The majority of builds and installations take place during night time, but there are some taking place also during day time.
- RPM installation times vary between 3-180 min and include waiting times
- In case of failures we end up with a huge backlog of many hours
- Combined with the other actions it is not the most efficient setup, thus a **parallel setup** is the best for us.



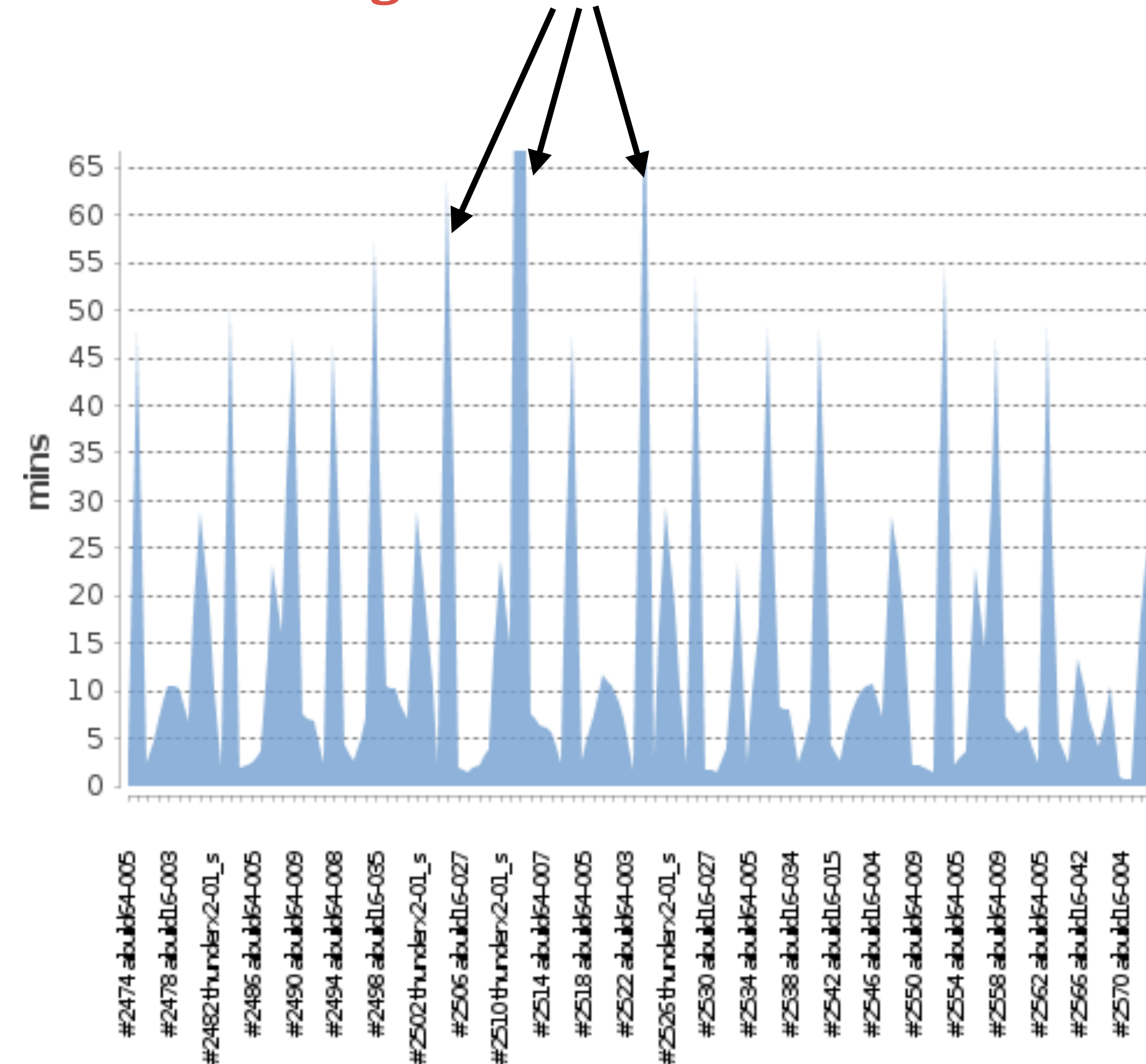
Parallel Installations

- We have been testing for a while parallel installations on atlas-nightlies-test.cern.ch
- Requires more than one release manager node
- Obvious solution: our build nodes can play that role and > 30 nodes were reconfigured as RMs which communicated with one gateway (gateway-cvmfs07.cern.ch)
- Everything that was installed on the production repo was also installed on the test repo for ca two months for testing
- We had to modify our installation scripts to cope with this setup:
 - creating robust **locks** in order to not start an installation while there was another one ongoing on a different release manager
 - easiest solution - a common /home directory on all build nodes where these locks are created
 - currently using the same code in production as for parallel testing

Parallel Installation Times

- The installation times range from 2 - 70 min depending on the size of the RPMs
 - the dbg and opt-dbg are much larger than opt
 - we remove older installations - aka “cleaning” of a directory path
- Once an installation is done, we need to update the AMI tags file with the latest build & installation information outside of the normal installation path
 - another lock is created immediately after an installation in order to update that file
- Tests are run after almost all installations, they need sometimes new ROOT files, checks done every two hours and updates performed if necessary - this part has not yet been tested, however they reside under their own path, no problems envisioned.

Large installations from scratch



Parallel Installations & CVMFS Actions

- The following actions need a transaction + publication at random times every day/night:
 - The nightly builds produce RPM files which are automatically installed - **no waiting time** 👍
 - ROOT files updates/deletions/additions for tests every two hours - **frequency to be increased** 🤔
 - One AMI tags file - updated after each RPM installation outside the installation path 🤔 🛠️
 - Removal of installations older than 30 days - **performed at the end of each installation** 👍
 - Installing privately build RPMs - when necessary 👍
 - Keep installations longer than 30 days - when necessary 👍
 - Development LCG nightly builds are installed on cvmfs as well 👍
 - Performing the garbage collection - daily/weekly? 🤔 Cronjob/Manual/...? 🤔

Parallel Installation Migration

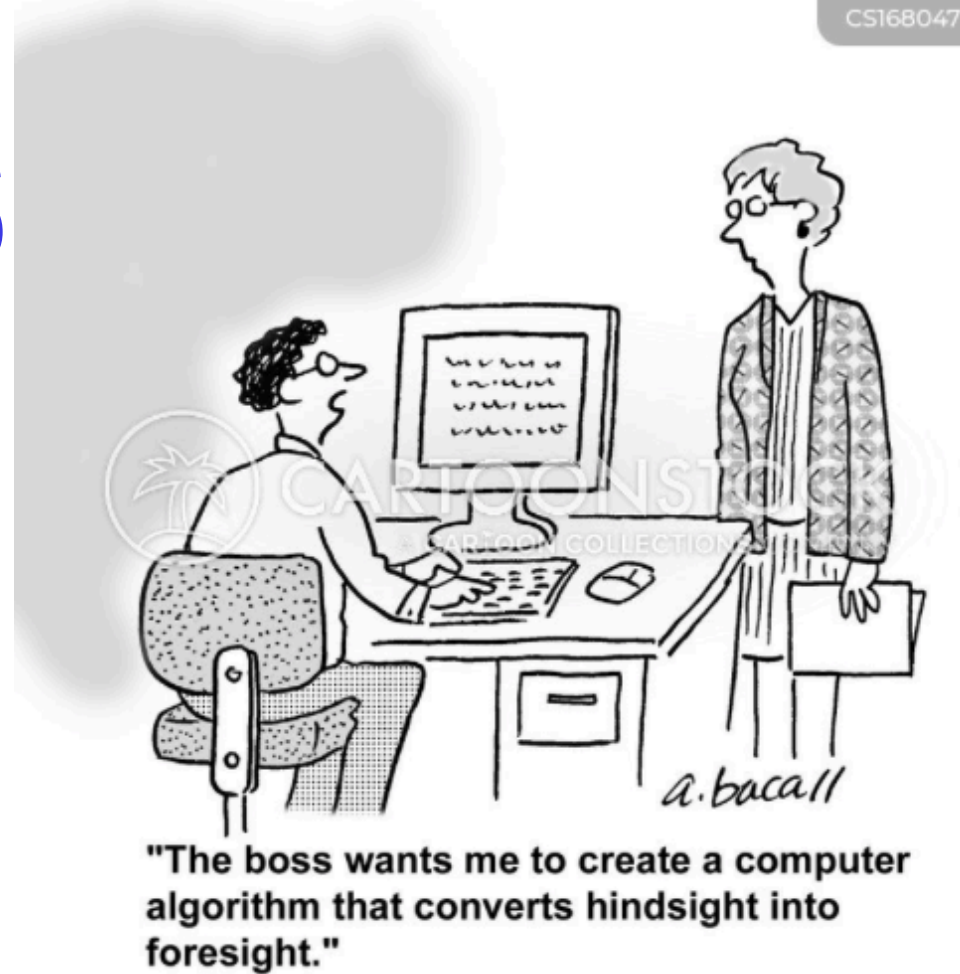
- Planned for the 26th of September
- Thanks in advance to Fabrizio for taking care of the necessary steps!
- Although our build nodes have been configured to talk to the (test) gateway which will be used for production as well, they will need to be reconfigured again for production mode
 - since they are used for our nightly and CI builds, need to coordinate when to reconfigure them after the migration (Chris Lee)
 - to start with, we will use the parallel setup with only one RM (our current publisher)
 - in ca 1-2 months (when all build nodes are reconfigured) we will use the full improvement range offered by the parallel system

Future Installation Software Improvements

- Split large installations (900+ packages, many GBs) into parts, publishing each part separately
- Investigate dnf5 package manager on CC9 ([Reiner Hauser](#))
 - it's goal is a new major version of libdnf, and a CLI also written in C++, plus almost all functionality of the old dnf tool. It is planned to be part of Fedora 38 (as an add-on), and meant to replace dnf in the future.
- Wrap the code and dependencies in a single binary for headache-free installation on the release managers (Brinick Simmons)
- moving from slc6 to CC7 our virtual environment needed extra library links, going to CS9 we will have similar issues



Desired CVMFS Improvements



- Initially the **AMI tags file** was updated on-the-fly whenever we installed and we didn't realise that it was not actually updating
 - there was no **warning** that we were actually **writing outside the opened transaction path**
- The **garbage collection**:
 - currently we can do this even if some installation is ongoing, which can lead to problems
 - would be great if the gateway would let us know that there is another transaction ongoing and either stop the garbage collection or **set a "in_transaction.lock"** file somewhere readable from all managers and let us create another type of lock for this action to be done in a safe mode

Conclusions

- CVMFS works very well and is stable
 - despite having many installations and other type of transactions daily we are coping well
 - the publication time has been improved since slc6
- parallel installations have been tested and migration imminent

Many thanks to Enrico - as well as to the entire CVMFS team - for the quick help in the past years and for always providing excellent user support! 😊