ALICE Offline services and operations

L. Betev, G. Eulisse, T. Wilken, M. Litmaath

Pre-GDB on operations, 24.02.2022

ALICE Offline operations

- A coin with two sides, each handled by its own team
  - What to run → build and CI operations (next pages)
  - Where to run → grid operations

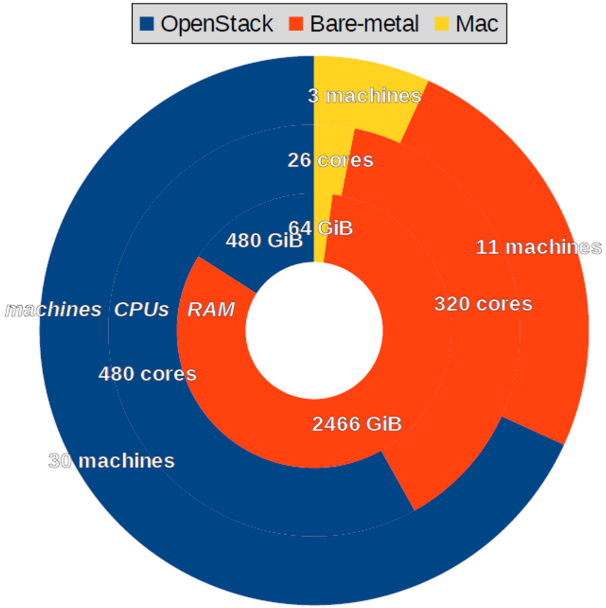
- Grid operations
  - Mostly managed through central services in the offline cluster
    - LDAP, CA, file catalog, task queue, job broker, job manager, MonALISA, …
  - MonALISA is crucial for many aspects
    - Monitoring of central services
    - Orchestration of productions, analysis trains and bulk data transfers
    - GUI for user job and data management
    - SE tests, to inform automatic and operator data management decisions
    - Tracking of grid site network metrics, ditto
    - Monitoring of site VOboxes, jobs, WN properties, SE disk servers
    - Accounting: VO perspective
    - And more…

WLCG connections

- VOMS, MyProxy, IAM
    - For VO management and pilot job submissions to CEs
- SAM
    - ETF only used for CE tests
    - SE and VObox test results are forwarded by MonALISA
- CRIC
    - SAM VO feed contents are determined from ALICE LDAP service
- Accounting: site perspectives
- Ops Coordination
    - Handling of matters not specific to ALICE, e.g. MW campaigns

ALICE Offline build and CI system

- 42 build machines with 826 CPU cores in total build pull requests for ALICE Offline repositories
  - – across different platforms (CentOS, *Ubuntu*, *OSX*) and architectures (x86, *arm64*) on OpenStack and the *offline cluster*
  - – using bespoke system of CI scripts + aliBuild as the build system
  - – for simpler tasks and code formatting: GitHub Actions
  - – 37 different checks across 13 different repositories
  - – 20–30 (non-trivial) builds + ~200 fast re-checks* per hour



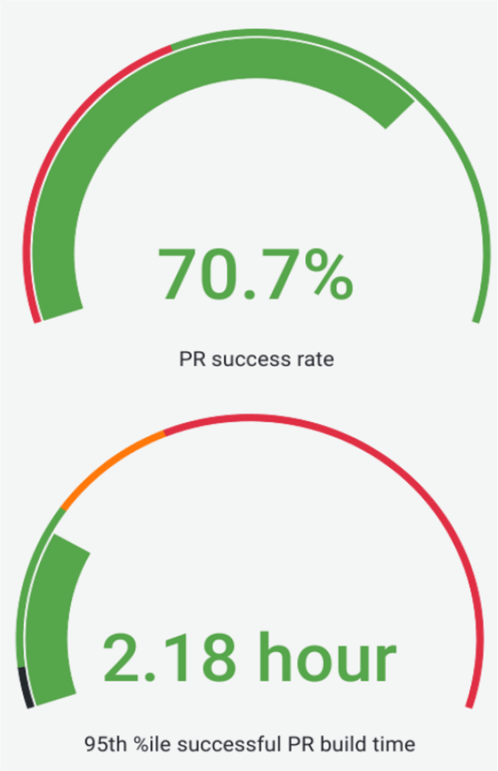
- Builds software every night, e.g. for use on the Grid, from CVMFS, as RPMs, etc

- Build processes scheduled using Mesos and Aurora on Linux, builds are containerised using Docker

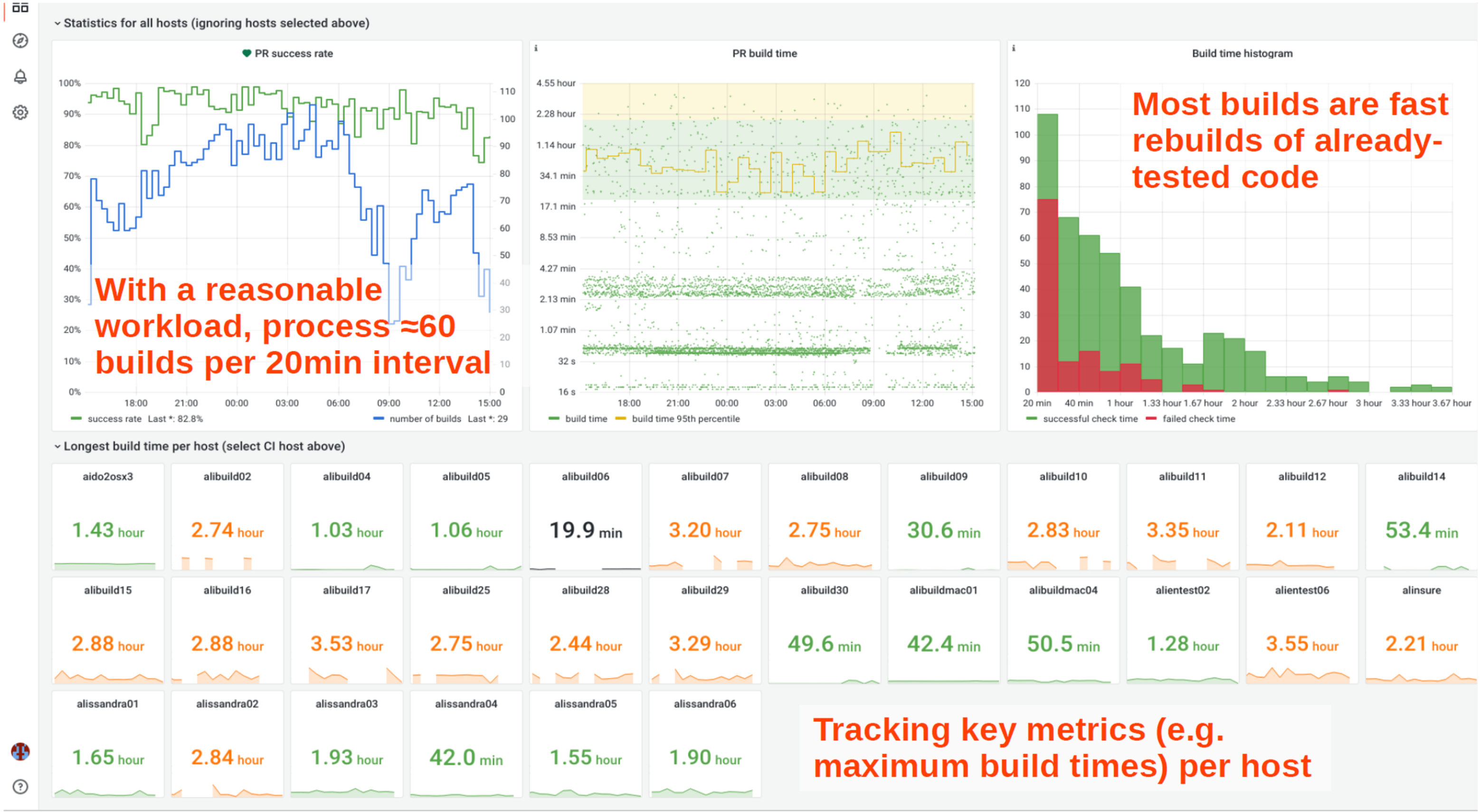
[*] re-builds of already tested code, to make sure it still builds against any code updated since its submission

CI system operation

- Scripts and configuration tracked using git
- GitLab used for some of the repositories
- Changes to CI scripts can be deployed on a rolling basis for testing
- OpenStack VMs managed through Puppet
- S3 for storing the build artefacts to use as a cache
- Bare-metal machines managed manually, as maintenance burden is fairly low
- Monitoring and alerting through Grafana



Monitoring



Statistics for all hosts (ignoring hosts selected above)

success rate Last *: 82.8%

number of builds Last *: 29

Longest build time per host (select CI host above)

aido2osx3	1.43 hour
alibuild02	2.74 hour
alibuild04	1.03 hour
alibuild05	1.06 hour
alibuild06	19.9 min
alibuild07	3.20 hour
alibuild08	2.75 hour
alibuild09	30.6 min
alibuild10	2.83 hour
alibuild11	3.35 hour
alibuild12	2.11 hour
alibuild14	53.4 min
alibuild15	2.88 hour
alibuild16	2.88 hour
alibuild17	3.53 hour
alibuild25	2.75 hour
alibuild28	2.44 hour
alibuild29	3.29 hour
alibuild30	49.6 min
alibuildmac01	42.4 min
alibuildmac04	50.5 min
alientest02	1.28 hour
alientest06	3.55 hour
alinsure	2.21 hour
alissandra01	1.65 hour
alissandra02	2.84 hour
alissandra03	1.93 hour
alissandra04	42.0 min
alissandra05	1.55 hour
alissandra06	1.90 hour

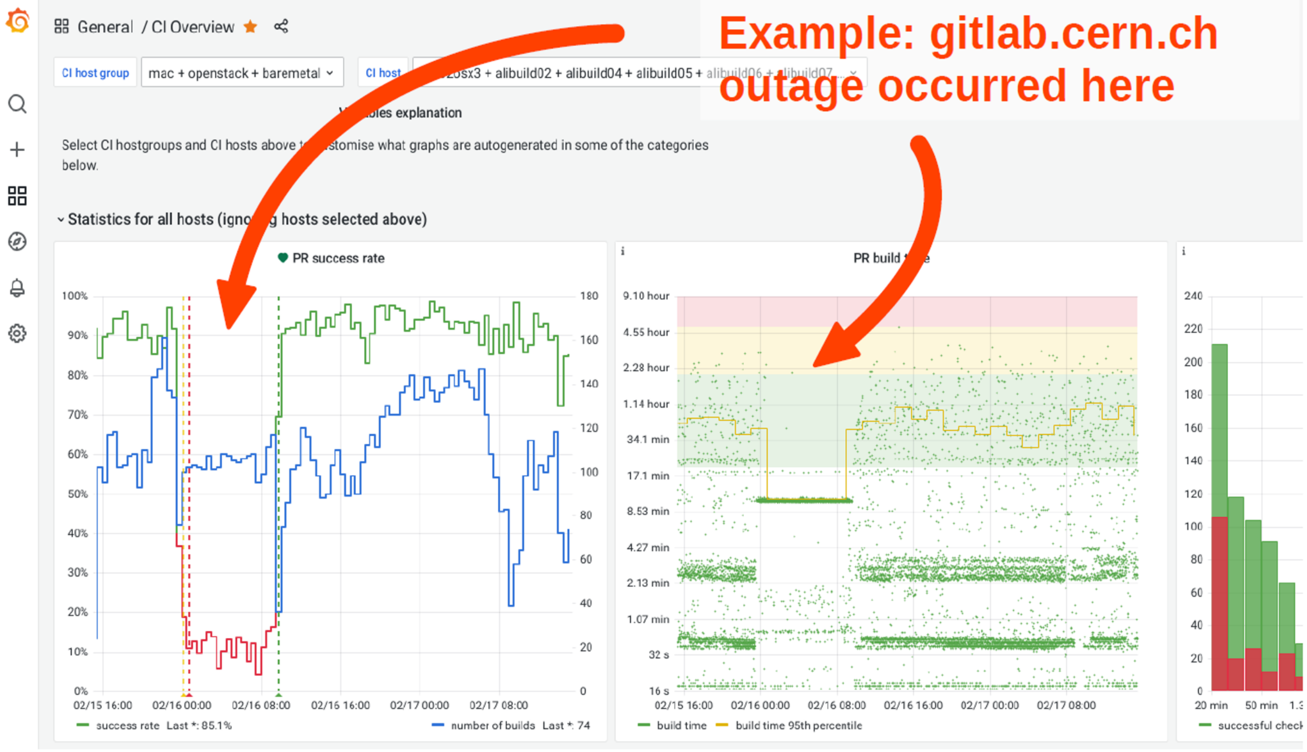
With a reasonable workload, process ≈60 builds per 20min interval

Most builds are fast rebuilds of already-tested code

Tracking key metrics (e.g. maximum build times) per host

CI monitoring: incident example

- CI system monitored using CERN IT administered Grafana instance
- Metrics pushed to CERN IT administered InfluxDB instance
- Alerting is easy to set up



Example: gitlab.cern.ch outage occurred here

General / CI Overview

CI host group: mac + openstack + baremetal

CI host: …20sx3 + alibuild02 + alibuild04 + alibuild05 + alibuild06 + alibuild07…

…bles explanation

Select CI hostgroups and CI hosts above t… …stomise what graphs are autogenerated in some of the categories below.

Statistics for all hosts (igno… …g hosts selected above)

Possible improvements

- On the ALICE side
    - Room for performance improvements
        - CI performance eaten up by stringent isolation of different pull requests from each other – there are likely ways to reduce this without compromising sandboxing
    - Unit testing coverage of some scripts can be improved, in order to make testing of simple changes to existing scripts simpler

- On the CERN IT side
    - Making the backend services **rock solid**, instead of providing more features
        - Cf. incidents affecting GitLab etc.
        - In other respects the CI system is rather low maintenance
    - Having macOS resources provided by CERN IT would be desirable