The ALICE Grid workflow for LHC Run 3
As deployed in production

Maxim Storetvedt | CHEP 2023 | Norfolk, VA | May 2023
Computing challenges in ALICE for Run 3

- Almost **10x** computing increase seen during Run 2
  - Average annual growth 15%
- ALICE detector, readout and software upgraded between 2018-2021
  - Increases the amount of collected data
  - From 4 GB/s to 100 GB/s post compression
- Number of jobs/pilots projected to increase
  - With more complex/multicore payloads
- Limitations to original AliEn Grid middleware stack
  - Maintenance & scaling concerns
    - Accumulation of dependencies and deprecated code
Changes for Run 3 and beyond

• New middleware introduced: JAliEn
  • New codebase
  • New backends
  • New features
• Updated means of deployment
  • More reliance on CVMFS
  • Introduction of containers
  • (More) automated steps
• Updated process for maintenance
  • Centrally triggered updates
  • Single recipe for key components
The new ALICE Grid workflow
The new ALICE Grid workflow

- **Containerised** core components
- Everything tied to a **central repository** and build system
  - Including the **payload environment**
  - Versions and changes are **automatic**
    - Pushed to CVMFS as needed
  - Essentials bootstrapped from **CVMFS**
    - Until we can get a container up and running
ALICE sites in Run 3

• ALICE/JAliEn still requires **VOBox** front-end for each site
  • Now available as a preconfigured **ready-to-go** container
  • Hosts site service component of JAliEn
• Container comes with **no** JAliEn binaries
  • All called directly from CVMFS as needed
  • When a job is matched, JAliEn VOBox component generates a pilot script
    • Points to both binaries and system libraries in CVMFS
    • Distributed to a free WN by site CE
• Only **two requirements** must be met to deploy a new JAliEn site
  • A resource management system (CE) for distributing pilot scripts across WNs
  • A valid host certificate
Job pilots and WNs

- Each startup script on WNs
  - Prepares environment
  - Loads pilot using libraries and Java from CVMFS
    - System agnostic
- Each JAliEn pilot consists of three components:
  - JAliEn JobRunner\(^1\): Resource/multicore handler
  - JAliEn JobAgent\(^2\): Job matcher/monitoring handler
  - JAliEn JobWrapper\(^2\): Payload executor
- The latter runs on a separate JVM for isolation
  - Automatically wrapped in a container by JobAgent
  - Handles payload that can be several cores per job slot
Payload environment

- By default, **all** Grid jobs are wrapped in a common **EL** container by JAliEn pilot
  - Provides a **tried-and-tested environment** on CentOS 7.9 across sites/nodes
  - Additional **isolation** from WN host
- Image as a sandbox directory located in CVMFS at
  - `/cvmfs/alice.cern.ch/containers/fs/singularity/centos-latest`
- Build recipe available on **Gitlab**
  - User PRs possible for package requests
- Two optional images can be set by **site**
  - **Alma 8.7**: For newer payloads (no ROOT5) and GPUs
  - **Alma 9.1**: Testing only (no production use)
- **GPUs are supported** through **Apptainer**
  - Compatibility check for supported container frameworks by JAliEn
  - GPUs auto detected, with flags/mounts added as needed
Compatibility

- Workflow only possible when several requirements met on WNs
  - E.g. OS, permissions, packages...
- Initially very low compatibility / only possible on a handful of sites
- Project started to check and map configurations across site WNs
  - **SiteSonar**: see presentation by Kalana Wijethunga
- Workarounds found as a result
  - Avoiding privileged bind-mounts through pre-created directories
  - Bundling needed OS **components** and **libraries** through CVMFS
    - Bootstraps placed for custom glibc and other required libraries
    - Everything needed to start both job **pilot** and a payload **container** provided by **CVMFS**
- Consequently, jobs can run on **any** WN with a recent Linux kernel and CVMFS
  - Roughly Linux 3.10 and later
Release distribution

• Each JAliEn release has a corresponding version tag in Alienv
  • Tool for tracking dependency trees and CVMFS paths for releases
  • Tightly interconnected with Alibuild - build system for new releases when tagged in Git
    • Successful builds automatically published to CVMFS
    • Also adds corresponding Alienv entry for newly built/published releases
• JAliEn may quickly be updated by applying a new Alienv tag
  • When done on a site VOBox, this will apply to all new job pilots
    • Through the startup scripts generated by JAliEn here
    • Full site is eventually switched with no further action needed
• Updates for VOBoxes triggered centrally across sites as new tags become available
On maintainability

• Updating JAliEn for site/WNs now largely automated (from site admin perspective)
  • Sites can subscribe to different release “channels”
    • **New** – Latest release in CVMFS
    • **Production** – Stable for general use
    • **Custom** – Specific version set by site
  • Published regularly (every ~2 weeks)
• Packages/environment now determined by containers independent of site
  • Common recipe for VOBox container
  • Common recipe for WN container
• JAliEn itself has also shown to be more maintainable
  • Several extensions since being brought into production

Compared to Run 2, a shift towards
• More steps being automated
• More steps managed centrally
Summary and outlook

- ALICE has moved to a new Grid middleware and workflow system based around JAliEn
  - Aimed at overcoming the computing challenges of LHC Run 3 and beyond
- Benefits from new developments in computing since release/creation of original AliEn
  - More reliance on CVMFS, giving more independence from host systems
  - Updated and more automated maintenance and deployment
  - Quick setup and more homogeneous environments through containers
- Streamlined codebase for better maintenance and further development
  - Removal of remaining AliEn legacy services
  - Support for WLCG tokens
  - Better utilisation of available Grid resources, through oversubscription
    - See next talk by Marta Bertran Ferrer
Thank You
[Questions, comments]?
email: mstoretv@cern.ch