Enabling Alternative Architectures

In the ALICE Computing Grid

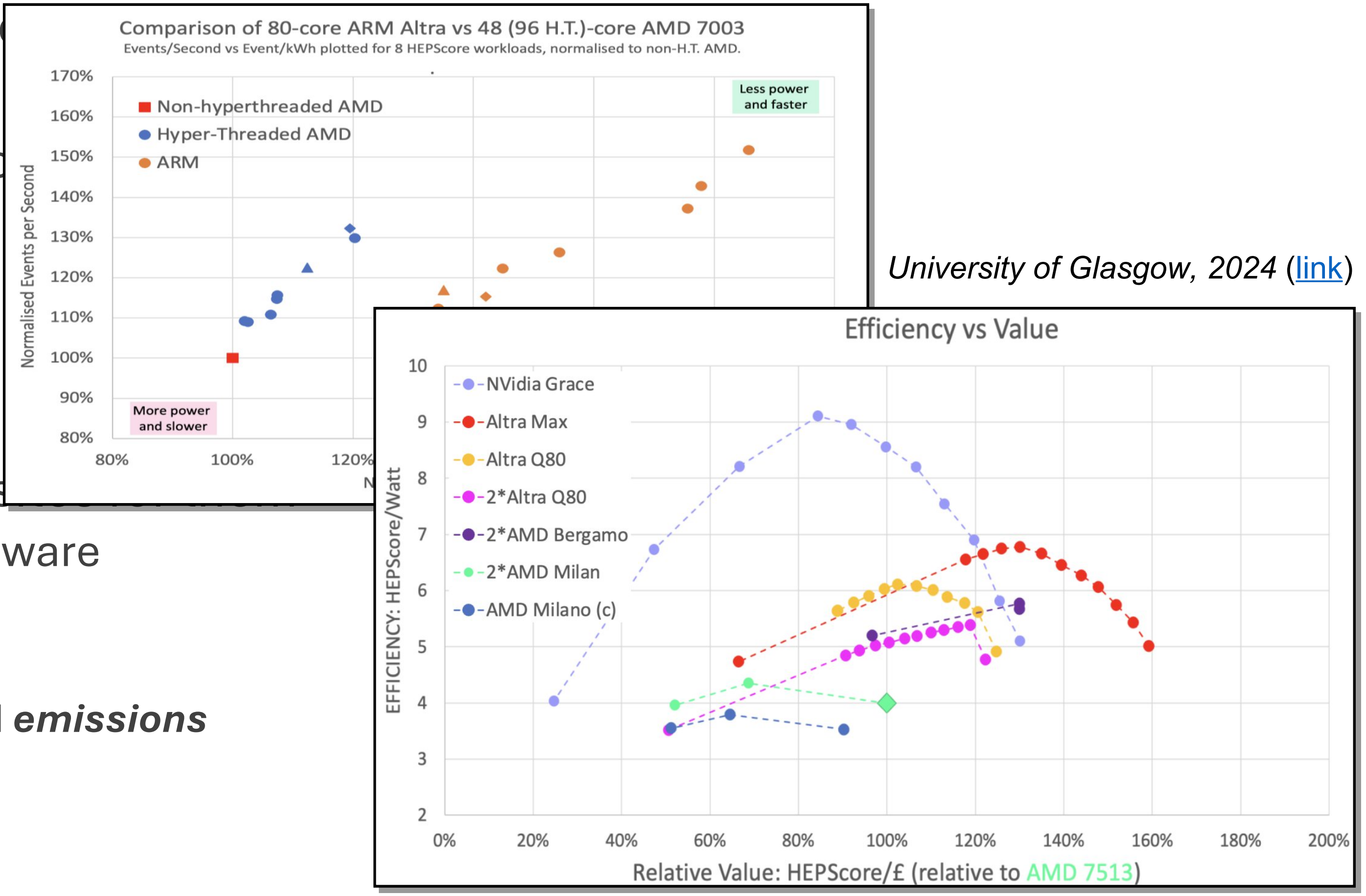
Maxim Storetvedt, *on behalf of the ALICE Collaboration* | CHEP 2024 | Kraków, PL | 24/10/2024

A changing resource landscape in the Grid

- Increasing availability of non-x86 based hosts in the WLCG
  - Especially those using **ARM**
- Can be attributed to wider selection of hardware options / OEMs...
  - Ampere Altra/One
  - Nvidia Grace
  - *Qualcomm SD1*
  - *Mediatek*
- ...but also increased interest among sites for them
  - Performance relative to price of hardware
  - Advertised with better **efficiency**
    - Possibility of lowering **energy** usage
    - Consequently cutting both **cost** and ***emissions***

A changing resource landscape in the Grid

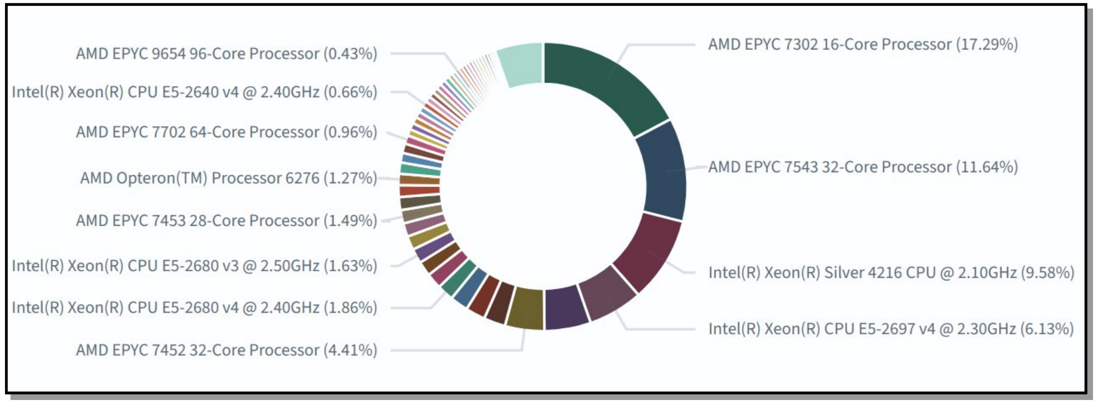
- Increasing availability of non-x86 bas
  - Especially those using **ARM**
- Can be attributed to wider selection o
  - Ampere Altra/One
  - Nvidia Grace
  - *Qualcomm SD1*
  - *Mediatek*
- ...but also increased interest among s
  - Performance relative to price of hardware
  - Advertised with better **efficiency**
    - Possibility of lowering **energy** usage
    - Consequently cutting both **cost** and ***emissions***



University of Glasgow, 2024 (link)

Status in the ALICE Grid

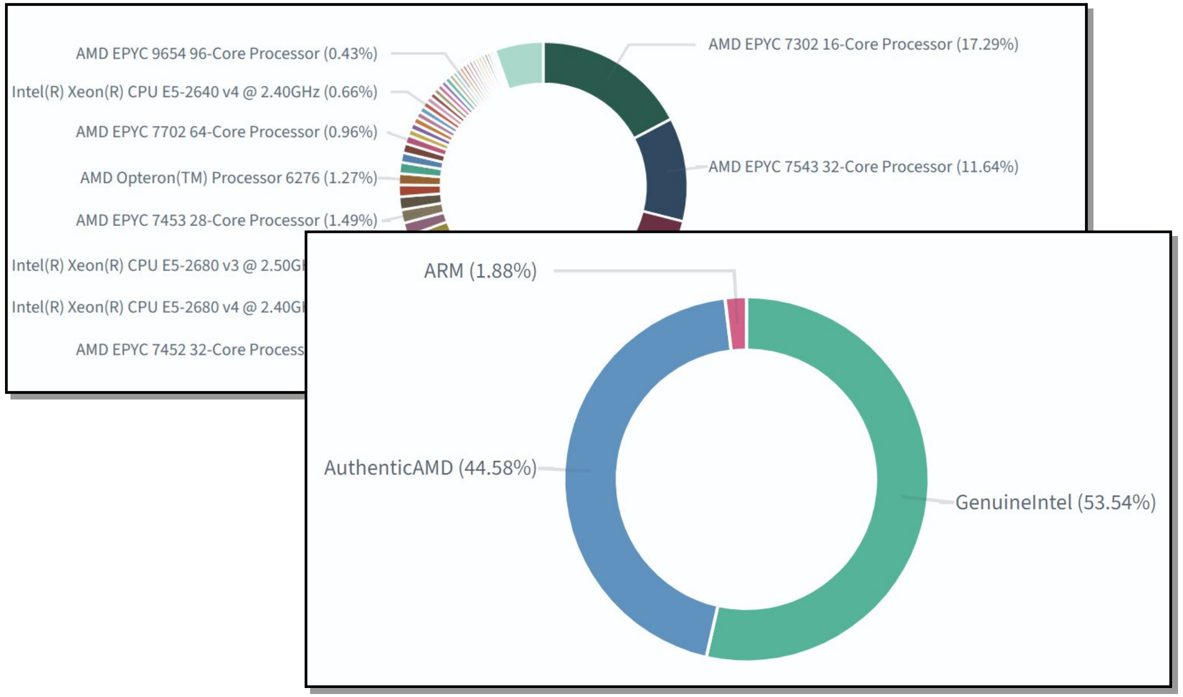
- ALICE subset of WLCG predominantly **x86_64**
    - With only x86 used for production workloads



Most common CPUs in ALICE Grid, Oct. 2024

Status in the ALICE Grid

- ALICE subset of WLCG predominantly **x86_64**
    - With only x86 used for production workloads
- But this is rapidly changing
    - Between Dec. 2023 - Oct. 2024
        - From 0 aarch64 cores to over **3000!**
    - More to come
        - Additional sites have expressed interest
- Must be ready to use all available resources!

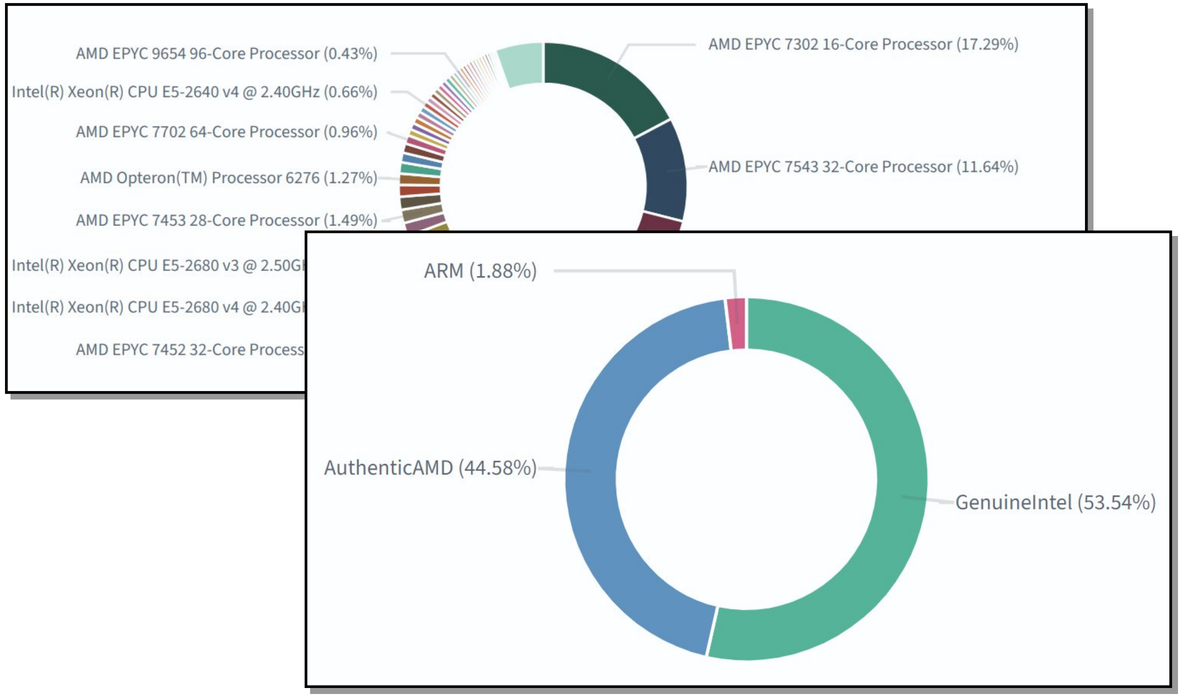


CPU vendors in ALICE Grid, Oct. 2024

Status in the ALICE Grid

- ALICE subset of WLCG predominantly **x86_64**
  - With only x86 used for production workloads
- But this is rapidly changing
  - Between Dec. 2023 - Oct. 2024
    - From 0 aarch64 cores to over **3000!**
  - More to come
    - Additional sites have expressed interest
- Must be ready to use all available resources!

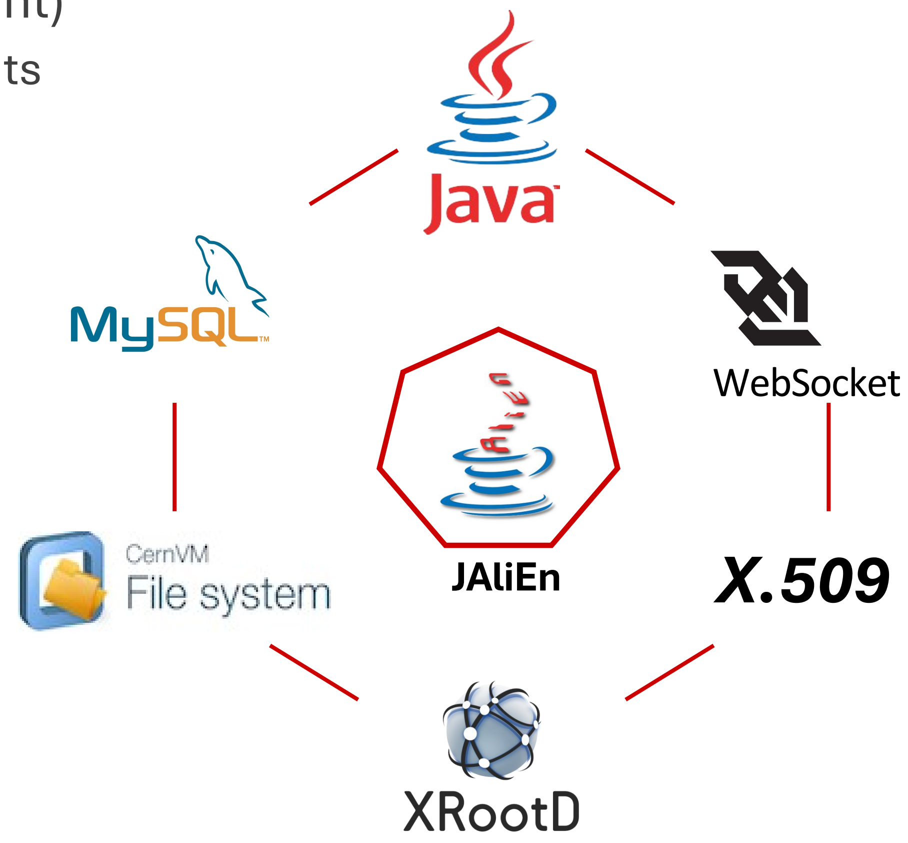
.... but can we include a different architecture *transparently*?



CPU vendors in ALICE Grid, Oct. 2024

The JAliEn middleware

- The ALICE Grid managed by **JAliEn** (Java **ALI**CE **En**vironment)
  - Grid middleware for site, central and user-facing components
- Benefits from the **portability** of Java…
- … but not everything is Java!
  - System binaries
  - Dependencies
  - Runtimes / containers
  - Job payloads
- Needs changes to
  - Accommodate middleware **binaries/dependencies**
  - Allow user jobs to be **matched** against aarch64 resources



Adding support for aarch64

- Initial aarch64 support added in **JAliEn 1.7.9** *(rel. Sept. 2023)*
- Since evolved to include
  - **Automatic** matching of **binaries**
  - **Automatic** matching of **jobs**
  - **Automatic** matching of **containers**
- Changes kept as generic as possible
  - Allows for simple slot-in of other architectures if needed
    - (More on this in a bit)
- End result allows aarch64 resources to be deployed and treated just **as any other** x86 host
  - Completely **transparent** for both **jobs** and **users!**

Ensuring compatibility

- Automatic matching of binaries
  - JAliEn is fully run from **CVMFS**
  - Binaries for each architecture can be provided in dedicated paths and builds

Ensuring compatibility

- Automatic matching of binaries
  - JAliEn is fully run from **CVMFS**
  - Binaries for each architecture can be provided in dedicated paths and builds
- Automatic matching of jobs
  - Once built, the **package** and its compatible ***platform*** is registered centrally
    - *Platform* is **OS version + architecture** of build (`el9-aarch64, el9-x86_64, el7-x86_64, etc.`)
  - WNs will advertise their platform, and be matched centrally against package availability

Ensuring compatibility

- Automatic matching of binaries
  - JAliEn is fully run from **CVMFS**
  - Binaries for each architecture can be provided in dedicated paths and builds
- Automatic matching of jobs
  - Once built, the **package** and its compatible ***platform*** is registered centrally
    - *Platform* is **OS version + architecture** of build (`el9-aarch64, el9-x86_64, el7-x86_64, etc.`)
  - WNs will advertise their platform, and be matched centrally against package availability
- Automatic matching of containers
  - Each WN can provide multiple (OS) platforms: attempts to find a **common build** OS across all the packages defined in a job
  - Combined with the system architecture to form a "*compatibility string*", which match package platforms
    - `e.g compat_el9-aarch64`
  - Each string corresponds to a CVMFS symlink to a compatible container
    - e.g: `compat_el9-aarch64 -> /cvmfs/alice.cern.ch/.../alma9-alice-20231212-aarch64`

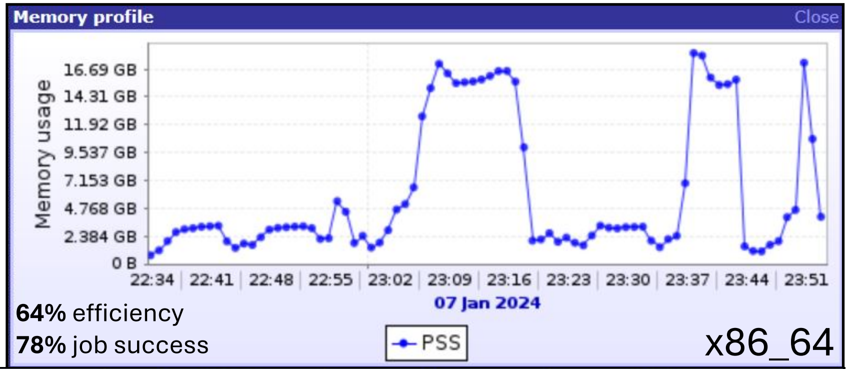
Ensuring compatibility (2)

- In other words: job matching is based on requested **packages**
  - Architecture of WN irrelevant as long as it can run all packages required by job
  - Compatible OS always provided by container
    - Availability of ***one*** aarch64 build for package across any OS is the only requirement for a job match
- Only requirement is ensuring that all packages and containers are built for **both** x86 and aarch64
- Dedicated build machine set up for this purpose
  - Ampere Altra Max 128 (N1) / 1TB RAM
  - Built packages automatically **registered** and pushed to **CVFMS**
    - Loaded from CVMFS by JAliEn using AliEnv (modulecmd wrapper) upon job start

Initial results

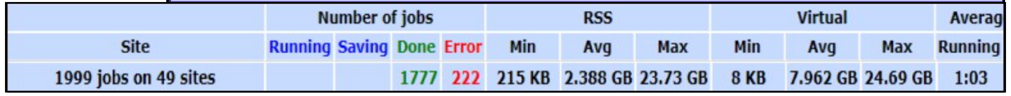
- First aarch64 resources available Dec.2023
  - Courtesy of University of Glasgow
  - Up and running within just a few days
    - No major issues
- Average x86 job was ~**22%** faster than aarch64[1]
  - And CPU efficiency **15%** higher
- But jobs on ARM have **99%** success rate!
  - Compared to **78%** on x86
  - No large memory spikes, which kill many x86 jobs
- **Conclusion:**
  - Aarch64 hardware is very promising for ALICE jobs
  - Lower cost than x86 alternatives also a bonus

NTIMEFRAMES = 5, NSIGEVENTS=200

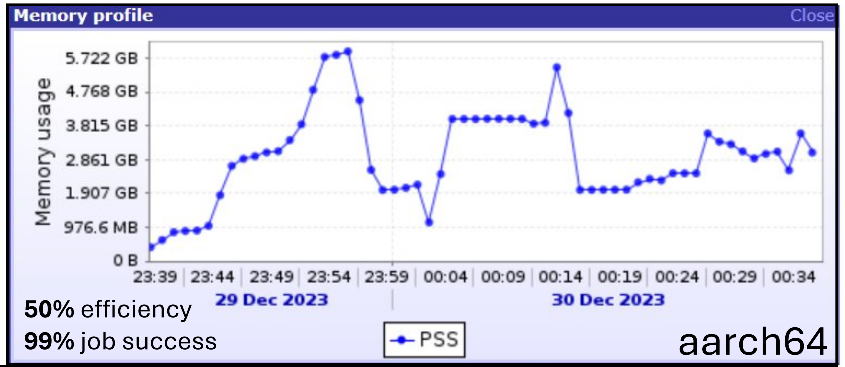


64% efficiency
78% job success

x86_64

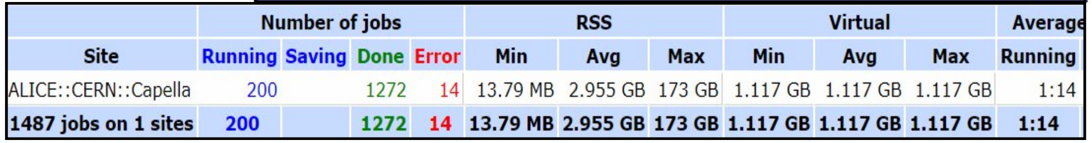


	Number of jobs				RSS			Virtual			Averag
Site	Running	Saving	Done	Error	Min	Avg	Max	Min	Avg	Max	Running
1999 jobs on 49 sites			1777	222	215 KB	2.388 GB	23.73 GB	8 KB	7.962 GB	24.69 GB	1:03



50% efficiency
99% job success

aarch64



	Number of jobs				RSS			Virtual			Average
Site	Running	Saving	Done	Error	Min	Avg	Max	Min	Avg	Max	Running
ALICE::CERN::Capella	200		1272	14	13.79 MB	2.955 GB	173 GB	1.117 GB	1.117 GB	1.117 GB	1:14
1487 jobs on 1 sites	**200**		**1272**	**14**	**13.79 MB**	**2.955 GB**	**173 GB**	**1.117 GB**	**1.117 GB**	**1.117 GB**	**1:14**

[1]: Ampere Altra Q80-30

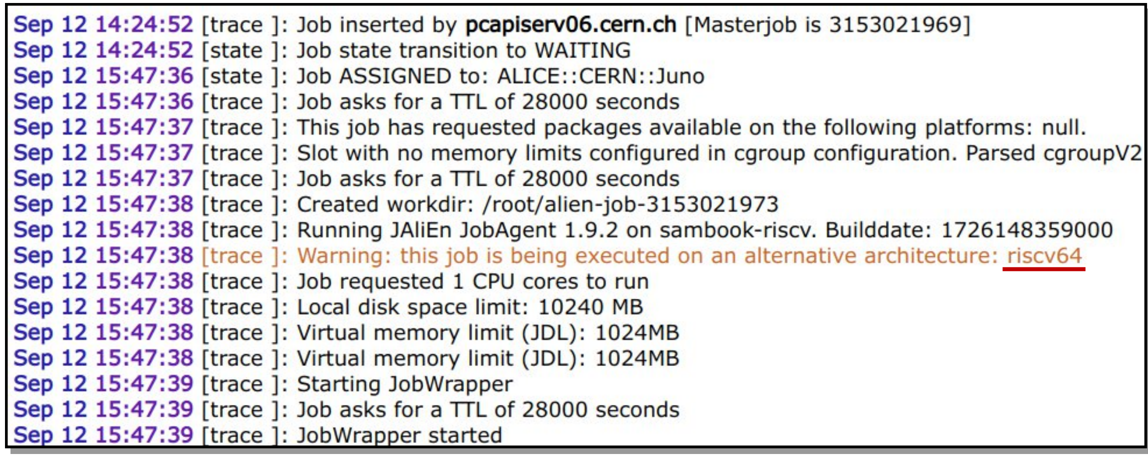
Caveat emptor

- Testing put on hold after Feb. 2024
    - Aarch64 jobs found to crash on a subset of kernels
        - Including the **default** kernel on Enterprise Linux 9 (EL9)
- Changing to a new architecture highlighted architecture-dependent code/assumptions
    - Including a possible memory overwrite, which was found and fixed [2]
    - Also present on x86, but behaviour between architectures very different
        - x86 would keep running, while aarch64 would crash immediately
- Delayed full adoption of aarch64 in production
    - But resulted in improved code quality / reliability across architectures
- Testing and physics validation to proceed once all looks good
    - Steps towards production use at ALICE in 2024!

Beyond ARM

- Adjustments to accommodate aarch64 not specific to just one architecture
  - Every change kept as generic as possible
- Simple to "slot-in" more architectures by
  - Adding appropriate binaries for JAliEn in dir for <arch> (in CVMFS)
    - e.g. /cvmfs/…/<arch>/JDK, /cvmfs/…/<arch>/apptainer
  - Creating <arch> versions of containers
  - Set up build machine to build/publish new packages
- Proof of concept: support added for `riscv64`
  - Riscv64 binaries for Java and Apptainer in CVMFS
  - Rest handled by JAliEn (binary / container matching)
    - Dependencies included in container image

Middleware components only, as there are no production packages built for riscv64



```
Sep 12 14:24:52 [trace ]: Job inserted by pcapiserv06.cern.ch [Masterjob is 3153021969]
Sep 12 14:24:52 [state ]: Job state transition to WAITING
Sep 12 15:47:36 [state ]: Job ASSIGNED to: ALICE::CERN::Juno
Sep 12 15:47:36 [trace ]: Job asks for a TTL of 28000 seconds
Sep 12 15:47:37 [trace ]: This job has requested packages available on the following platforms: null.
Sep 12 15:47:37 [trace ]: Slot with no memory limits configured in cgroup configuration. Parsed cgroupV2
Sep 12 15:47:37 [trace ]: Job asks for a TTL of 28000 seconds
Sep 12 15:47:38 [trace ]: Created workdir: /root/alien-job-3153021973
Sep 12 15:47:38 [trace ]: Running JAliEn JobAgent 1.9.2 on sambook-riscv. Builddate: 1726148359000
Sep 12 15:47:38 [trace ]: Warning: this job is being executed on an alternative architecture: riscv64
Sep 12 15:47:38 [trace ]: Job requested 1 CPU cores to run
Sep 12 15:47:38 [trace ]: Local disk space limit: 10240 MB
Sep 12 15:47:38 [trace ]: Virtual memory limit (JDL): 1024MB
Sep 12 15:47:38 [trace ]: Virtual memory limit (JDL): 1024MB
Sep 12 15:47:39 [trace ]: Starting JobWrapper
Sep 12 15:47:39 [trace ]: Job asks for a TTL of 28000 seconds
Sep 12 15:47:39 [trace ]: JobWrapper started
```

Riscv64 job running in JAliEn 1.9.2

Summary and outlook

- **ARM** and other non-x86 resources are becoming increasingly more **relevant** in the ALICE Grid
  - Increased hardware **availability** and **competitiveness**
  - Important to ensure these new resources can be fully **utilised**
- Support for aarch64 added within the ALICE middleware **JAliEn**
  - Use of x86/aarch64 resources completely **transparent** for both jobs and end users
  - Achieved by matching jobs by the **packages** available across WNs
- Provided changes can also scale across **multiple architectures**
  - Adding Java + container binaries to CVMFS is enough
    - Rest handled by JAliEn
  - Proof-of-concept **riscv64** support already enabled
- Initial experience from running ALICE Grid jobs on aarch64 hardware very **promising**
  - Delay due to necessity to validate the experimental software and fix platform-exposed bugs
  - Ready to resume testing at **full speed** once fix is merged
    - Towards production!