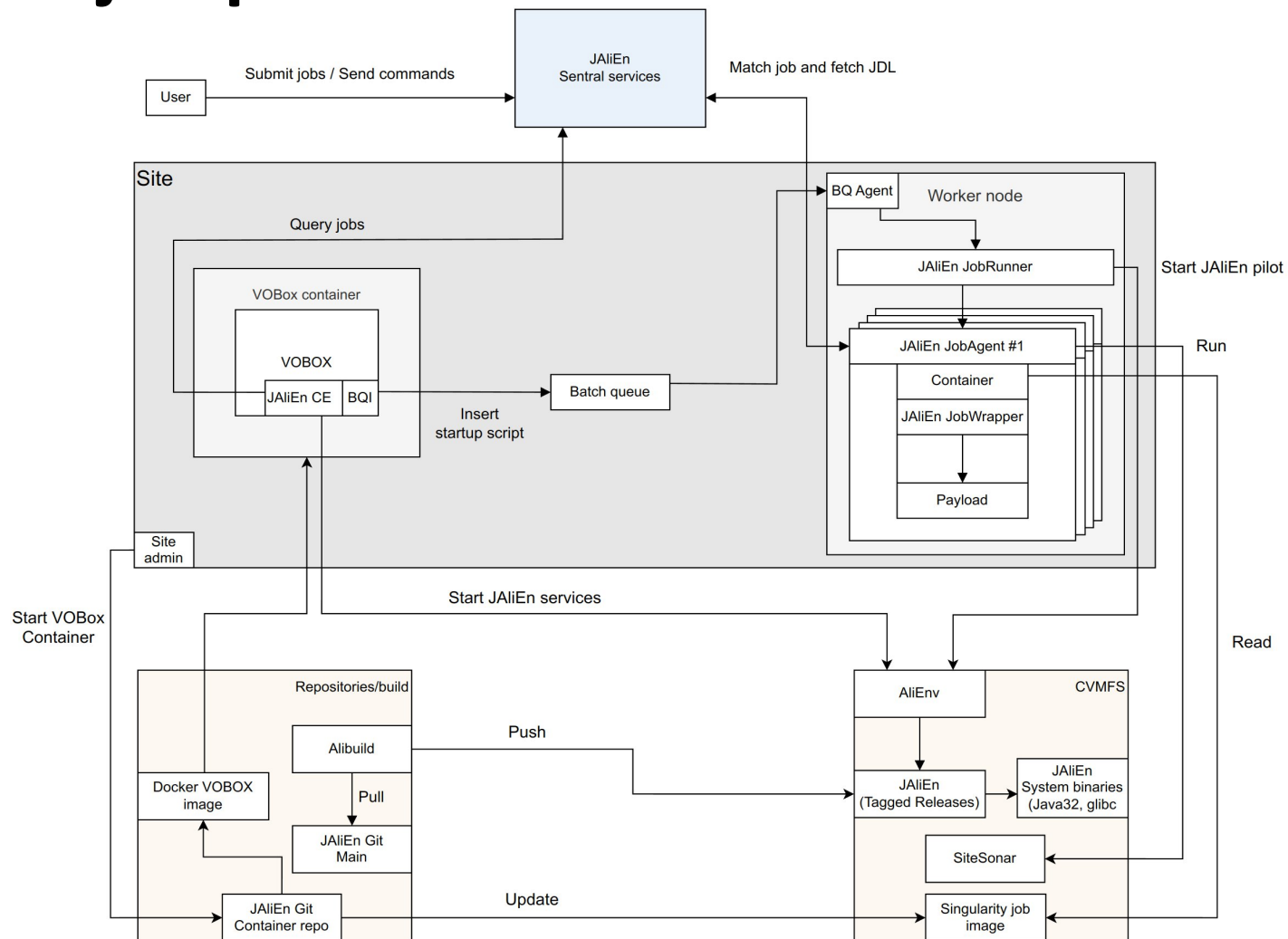


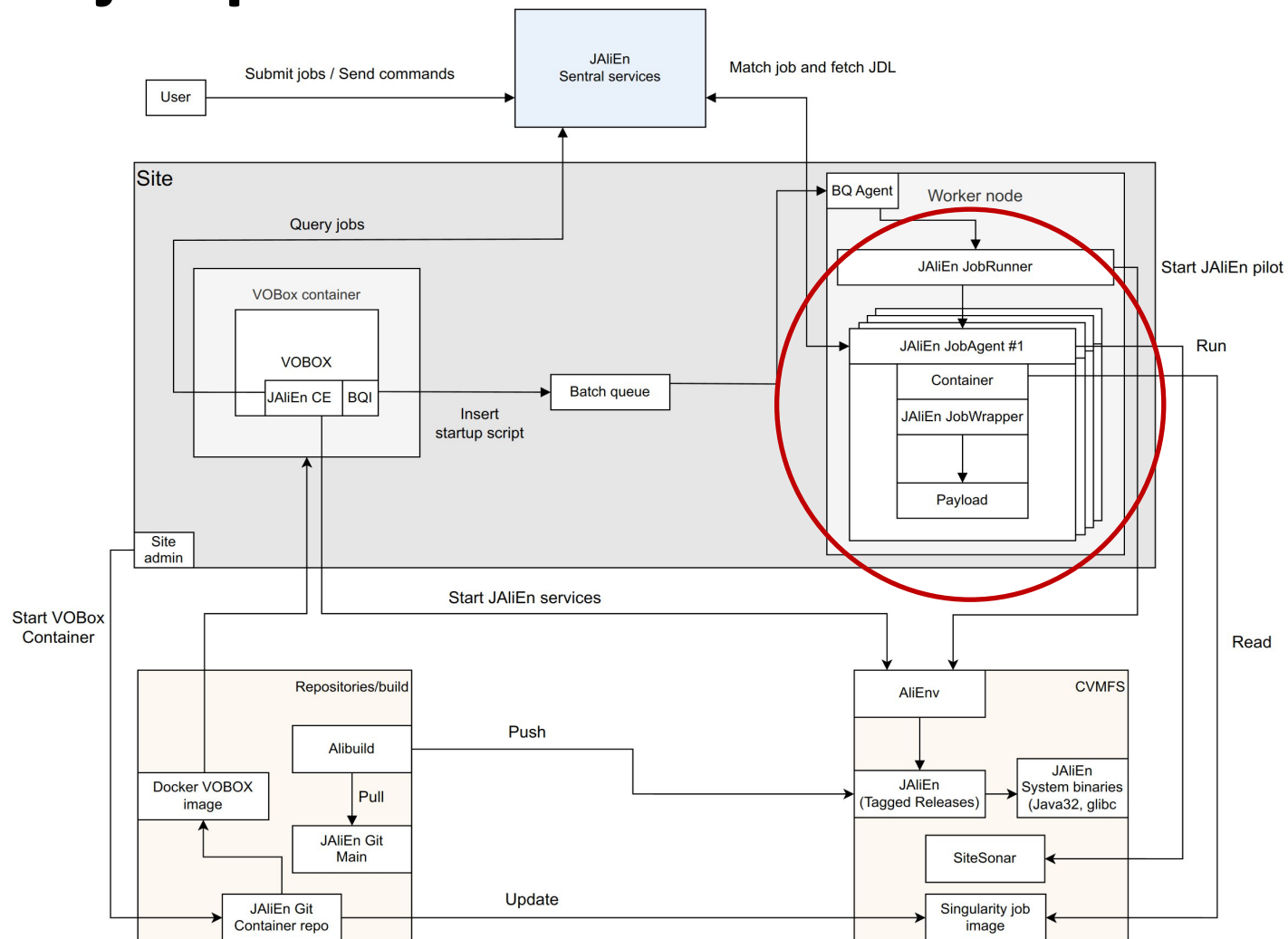


Job pilot features and job isolation

Background: job pilots and WNs

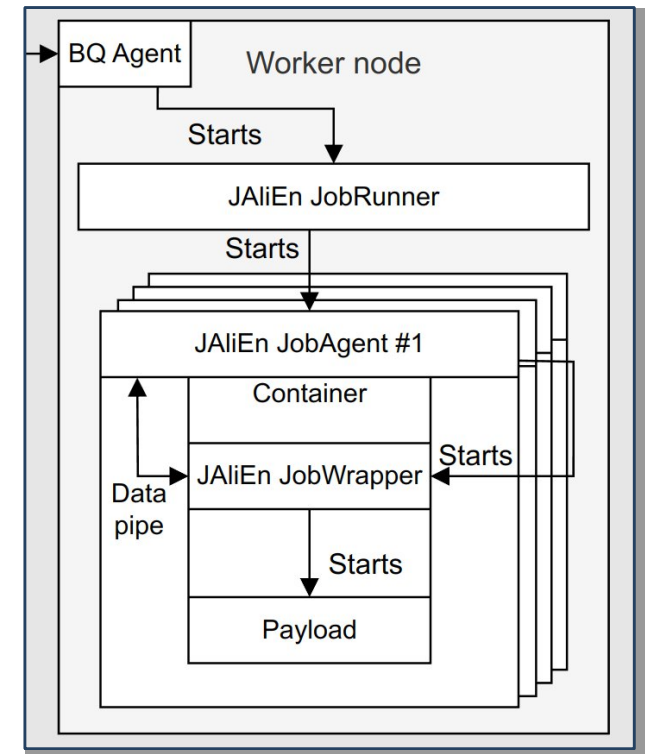


Background: job pilots and WNs



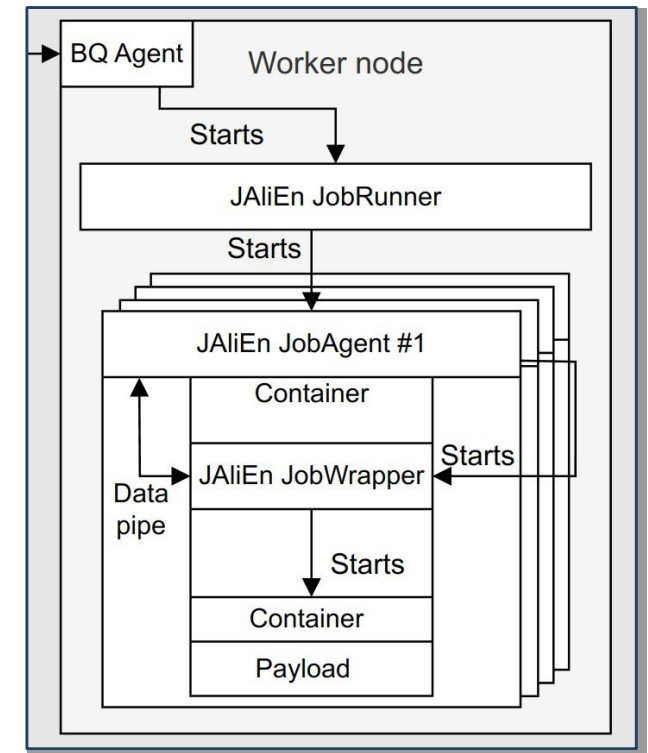
Job pilots and WNs

- 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
- Started by script generated by JAliEn CE, which
 - Prepares environment
 - Loads pilot using libraries and Java from CVMFS
 - System agnostic
- The JobWrapper runs on a separate JVM
 - Handles payload that can be several cores per job slot
 - Automatically **put in an isolated container** by JobAgent



Job pilots and WNs

- 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
- Started by script generated by JAliEn CE, which
 - Prepares environment
 - Loads pilot using libraries and Java from CVMFS
 - System agnostic
- The JobWrapper runs on a separate JVM
 - Handles payload that can be several cores per job slot
 - Automatically **put in an isolated container** by JobAgent
 - When possible, **also wraps the payload in a container** for process control



Job containers & environment

- **All** Grid jobs **must** now be wrapped by a top-level container
 - Provides a **tried-and-tested environment** across sites/nodes
 - Additional **isolation** from WN host
- Images located located in CVMFS
 - `/cvmfs/alice.cern.ch/containers/fs/singularity`
 - Build recipe available on [Gitlab](#)
 - PRs possible for package requests
- Specific image **selected by JAliEn** based on required packages for job
 - Can be CentOS 7, Alma 8 or Alma 9
 - Override possible by site in “*container.properties*”
- When **multiple** containers used, the **same image** is propagated down
- GPUs are **supported** across the multiple layers
 - Compatibility check done by JAliEn, with flags/mounts added as needed

Notable job-pilot changes

- Re-enabled **idle timeout** for jobs
 - Jobs that do nothing for longer durations will be **killed**
 - I.e. **never** once surpassed **10% cputime within 15 min**
- All WNs without AVX support are now **blocked** from production jobs
- Failed jobs that never reach a RUNNING state are automatically **resubmitted**
 - But only up to **three times**
- New Job states
 - **ERROR_A**: Job matched, but unable to continue
 - **ERROR_VN**: Validation failed to start
- Option provided to **skip** check for **corefiles**
- Option provided to manually set WN **architecture**
- ***Alternative architectures***
- ***Cgroups v2 for enforcing resource constraints***



Alternative architectures

Alternative architectures

- **Aarch64 supported** as of JAliEn **1.7.9**
 - Hardcoded x86 paths removed
 - **Automatic** matching of **binaries**
 - JAliEn CE and job-pilots
 - **Automatic** matching of **containers**
 - Corresponding aarch64 versions of platforms requested by job
 - Monitoring adjusted to work across architectures
 - Updated ML script for aarch64 **voboxes**
- Changes kept as generic as possible
 - Allows us to easily slot-in support for more architectures in future (**e.g. RISC-V**)



Alternative architectures

- **Aarch64 supported** as of JAliEn **1.7.9**
 - Hardcoded x86 paths removed
 - **Automatic** matching of **binaries**
 - JAliEn CE and job-pilots
 - **Automatic** matching of **containers**
 - Corresponding aarch64 versions of platforms requested by job
 - Monitoring adjusted to work across architectures
 - Updated ML script for aarch64 **voboxes**
- Changes kept as generic as possible
 - Allows us to easily slot-in support for more architectures in future (**e.g. RISC-V**)
- But, for now, jobs with aarch64 builds must **manually** be submitted to aarch64 sites
 - To be changed with new brokering service by Kalana Wijethunga ([link](#))
 - Allows aarch64 sites to transparently run and match jobs just as any other x86 site
 - Enabled by new Optimizer by Haakon André Reme-Ness ([link](#))





Improving job isolation with Cgroups v2

Improving job isolation with Cgroups v2

- **Cgroups v2** allows for **fine-grained resource control** per job
 - Controllers for CPU, memory, IO...
 - In theory, can be used **unprivileged** – perfect for integrating with job pilot
 - Can be automatically handled by Apptainer, but...
- **...unprivileged Cgroups v2** forced to run on **user.slice***
 - Apptainer insists on moving its processes there
 - **Outside** of original slot cgroup!
 - And only possible for an interactive user with a session
 - Or, if **lingering** for a user is enabled
 - Must be done in advance by site admin

Improving job isolation with Cgroups v2

- **Cgroups v2** allows for **fine-grained resource control** per job
 - Controllers for CPU, memory, IO...
 - In theory, can be used **unprivileged** – perfect for integrating with job pilot
 - Can be automatically handled by Apptainer, but...
- **...unprivileged Cgroups v2** forced to run on **user.slice***
 - Apptainer insists on moving its processes there
 - **Outside** of original slot cgroup!
 - And only possible for an interactive user with a session
 - Or, if **lingering** for a user is enabled
 - Must be done in advance by site admin

However,
moving outside of the slot cgroup
also means **no more slot limits**

Working around unprivileged limitations

- Fully unprivileged cgroups generally use **user.slice**
 - Set up automatically upon user login
- **But**, other cgroups *can* be handed to unprivileged users
 - If prepared in advance by a privileged user
- Specifically, a user must be given ownership of
 - The **cgroup** — i.e. its top-level directory in `/sys/fs/cgroup`
 - The **cgroup.procs** file — to allow moving processes in/out of it
 - The **cgroup.subtree_control** file — to allow delegation of controllers to subgroups

...but should this not already done by CE/LRMS when slot cgroup is constructed?

Status of unprivileged cgroups v2 in common CEs

- Behaviour of both SLURM and HTCondor checked, i.e.
 - How cgroup is created for each slot
 - What ownership/permissions given
- **SLURM**
 - New cgroup dir created on each new job for slot
 - Cgroup ownership set to that of the executing user — **Great!**
 - ...but **only** cgroup ownership. All files inside still owned by root
 - Can unprivileged create new cgroups, but not move anything there
 - No unprivileged delegation of controllers
- **HTCondor**
 - New cgroup dir created on each new job for slot
 - But **all** files/directories owned by root



Workaround: a custom cgroups v2 plugin (1)

- Proof of concept Cgroups v2 plugin created for **SLURM (22.05)**
 - Existing setup already “halfway there”
 - Simple to extend functionality
 - Sets the appropriate permissions, and checks for subgroups in cleanup
- Upon building the cgroup
 - Check for given privileges
 - Attempt creating subgroups, move processes and apply limits
 - Attempt breaking limit



<https://github.com/SchedMD/slurm/compare/slurm-22.05...zensanp:slurm:slurm-22.05>

Workaround: a custom cgroups v2 plugin (1)

```
[root@alieevee-wn-5 user]# pwd
/sys/fs/cgroup/system.slice/slurmstepd.scope/job_214324/step_0/user
[root@alieevee-wn-5 user]# ls -ltr
total 0
-rw-r--r--. 1 root root 0 Nov 9 17:06 memory.max
-rw-r--r--. 1 root root 0 Nov 9 17:06 memory.high
-rw-r--r--. 1 root root 0 Nov 9 17:06 cgroup.subtree_control
-rw-r--r--. 1 root root 0 Nov 9 17:06 cgroup.procs
drwxr-xr-x. 4 runner runner 0 Nov 9 17:06 tasek_0
-r--r--r--. 1 root root 0 Nov 9 17:12 memory.swap.events
-r--r--r--. 1 root root 0 Nov 9 17:12 memory.events
-rw-r--r--. 1 root root 0 Nov 9 17:12 cgroup.freeze
--w-----. 1 root
-rw-r--r--. 1 root
-rw-r--r--. 1 root
```

```
[root@alieevee-wn-5 user]# cd tasek_0/
[root@alieevee-wn-5 tasek_0]# ls -ltr
total 0
-r--r--r--. 1 root root 0 Nov 9 17:06 memory.swap.current
-r--r--r--. 1 root root 0 Nov 9 17:06 memory.stat
-r--r--r--. 1 root root 0 Nov 9 17:06 cpu.stat
-rw-r--r--. 1 runner runner 0 Nov 9 17:06 cgroup.subtree_control
-rw-r--r--. 1 runner runner 0 Nov 9 17:06 cgroup.procs
-rw-r--r--. 1 root root 0 Nov 9 17:06 memory.max
-rw-r--r--. 1 root root 0 Nov 9 17:06 cpu.max
drwxrwxr-x. 2 runner runner 0 Nov 9 17:06 runner
drwxrwxr-x. 2 runner runner 0 Nov 9 17:06 agents
--w-----. 1 root root 0 Nov 9 17:12 cgroup.kill
-rw-r--r--. 1 root root 0 Nov 9 20:39 memory.swap.max
-rw-r--r--. 1 root root 0 Nov 9 20:39 memory.swap.high
```

Upstreaming?

- Proof-of-concept **working** as expected
 - Tested on the *Eevee* cluster (ALICE::CERN::Eevee)
- But changes only useful if they can be **upstreamed**
 - Unlikely sites willing install external plugins/rpms
 - Unofficial changes prohibited on sites subscribed to SLURM paid support
- Request sent via the **slurm-users** mailing list...
 - No permissions to do PRs / report issues on public repository
 - No other official channels (aside from paid)
 - Archive: <https://groups.google.com/g/slurm-users/c/v7e5DSUvPPw>
- ... but **no response**

A custom cgroups v2 plugin (2)

- Proof-of-concept Cgroups v2 features added for **HTCondor (v23)**
 - Similar as for SLURM
 - Sets the appropriate permissions/ownership when building the group
- As before
 - Check for given privileges
 - Attempt creating subgroups, move processes and apply limits
 - Attempt breaking limit



<https://github.com/htcondor/htcondor/compare/main...zensanp:htcondor:main>

Upstreaming!

- Proof-of-concept **working** on HTCondor as for SLURM
 - Also tested on Eevee as before
- Likewise, only useful if same changes can be added **upstream**
 - Availability to increase as sites upgrade their versions
- Request sent via the **condor-users** mailing list...
 - ...and almost immediate response!
 - Interest from Condor devs
 - Experimented with this earlier, but not quite working
 - Potential use-case with GlideInWMS
 - Included in test branch within 2 weeks
 - Big thanks to **Greg Thain!**
 - ~~Still some missing features*~~

Upstreaming!

- Proof-of-concept **working** on HTCondor as for SLURM
 - Also tested on Eevee as before
- Likewise, only useful if same changes can be added **upstream**
 - Availability to increase as sites upgrade their versions
- Request sent via the **condor-users** mailing list...
 - ...and almost immediate response!
 - Interest from Condor devs
 - Experimented with this earlier, but not quite working
 - Potential use-case with GlideInWMS
 - Included in test branch within 2 weeks
 - Big thanks to **Greg Thain!**
 - ~~Still some missing features*~~

Merged to main and included
in HTCondor 23.1!

New cgroups v2 features within the job pilot

- Previous cgroups v2 implementation used containers/containerizer
 - Built-in features of Apptainer
- Not compatible with the cgroup given by new HTCondor/SLURM
 - Apptainer always insists on moving processes
- Cgroup setup and management must now be done **manually** by **the job pilot**
- Logic adjusted to accommodate the changes:
 - Top level slot cgroup creation and delegation of controllers
 - Via **JobRunner**
 - Job cgroup creation and limits
 - Via **JobAgent**



Constructing cgroup in JR/JA

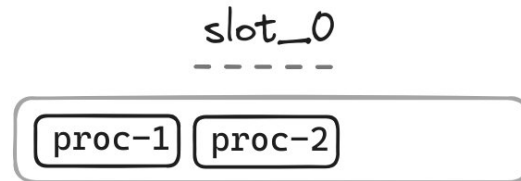
Constructing cgroup in JR/JA



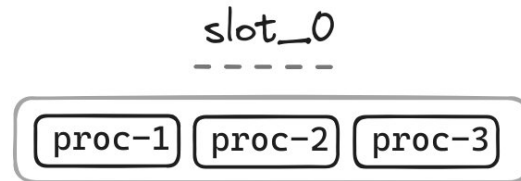
Constructing cgroup in JR/JA



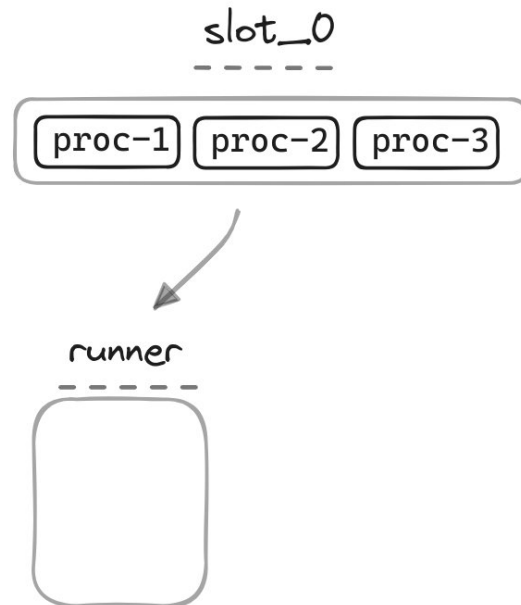
Constructing cgroup in JR/JA



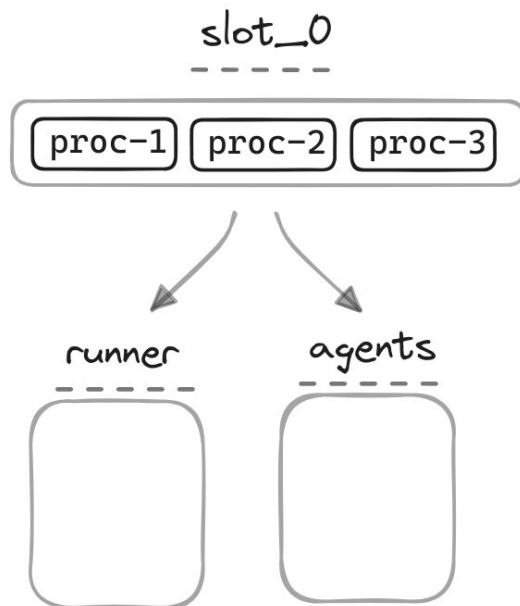
Constructing cgroup in JR/JA



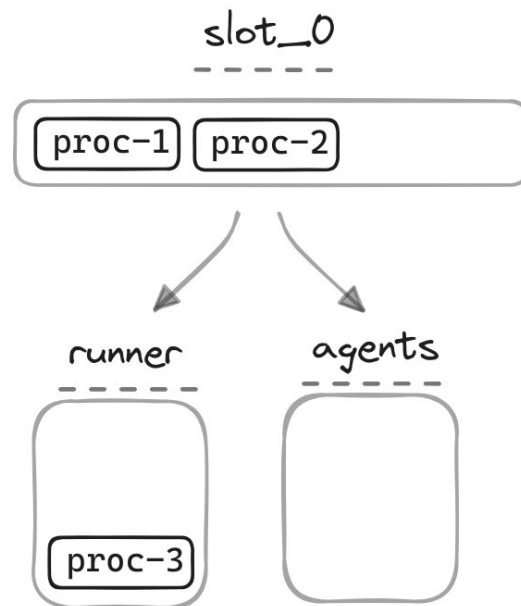
Constructing cgroup in JR/JA



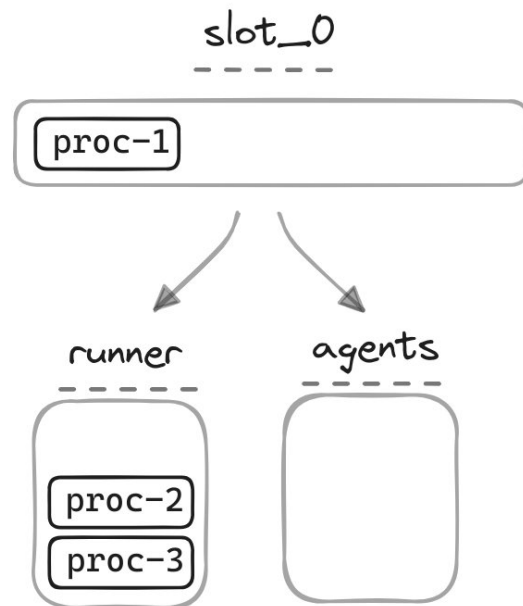
Constructing cgroup in JR/JA



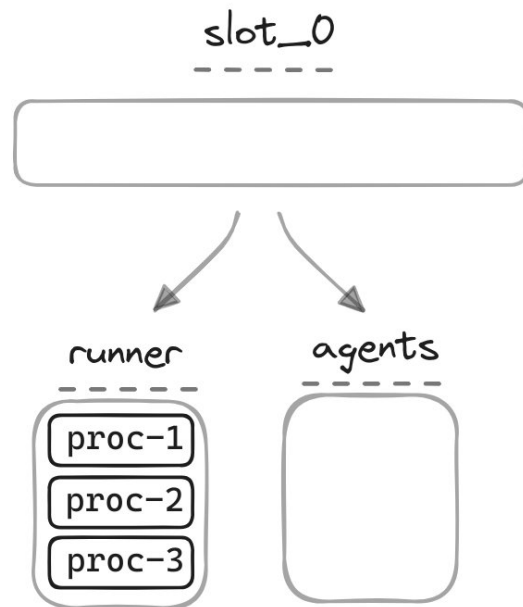
Constructing cgroup in JR/JA



Constructing cgroup in JR/JA



Constructing cgroup in JR/JA

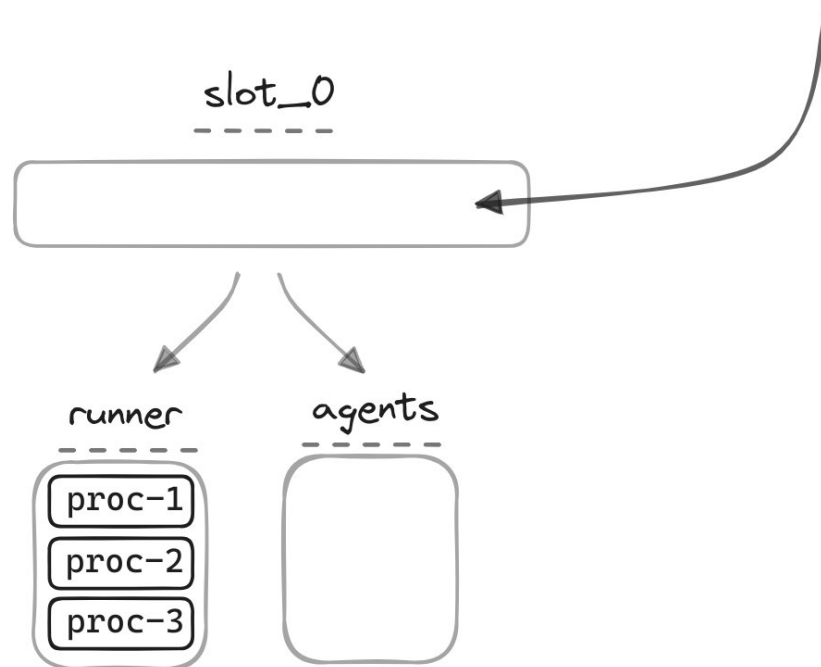




ALICE

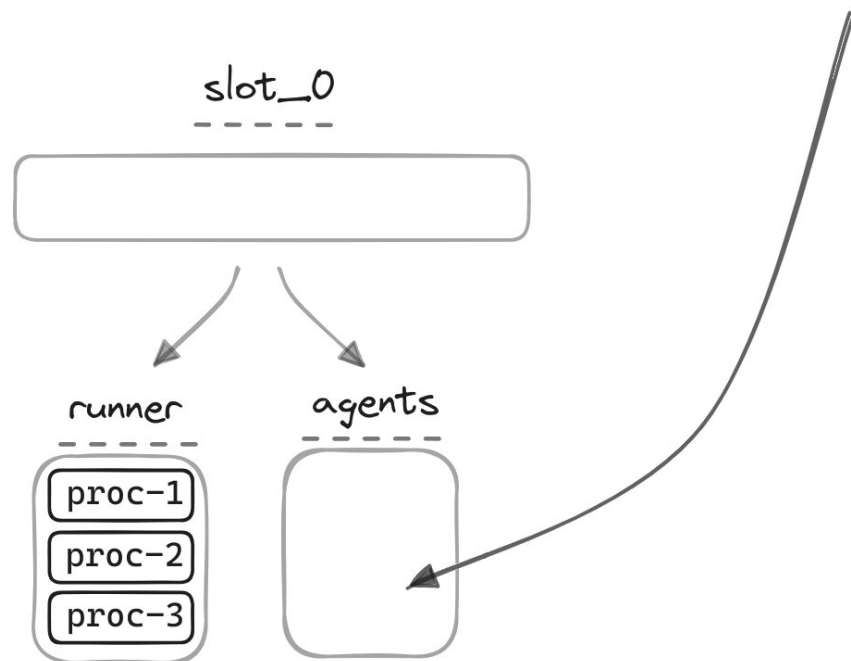
Constructing cgroup in JR/JA

Delegated controllers
(via `cgroup.subtree_control`)

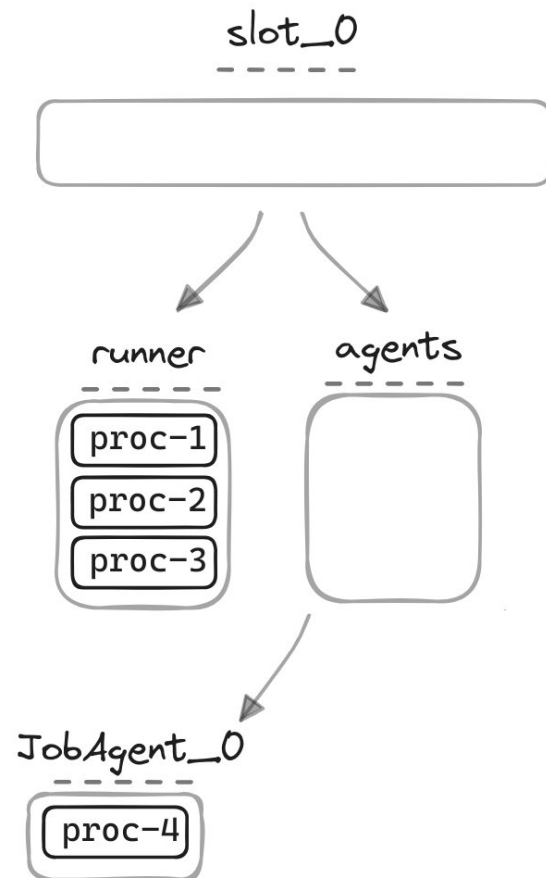


Constructing cgroup in JR/JA

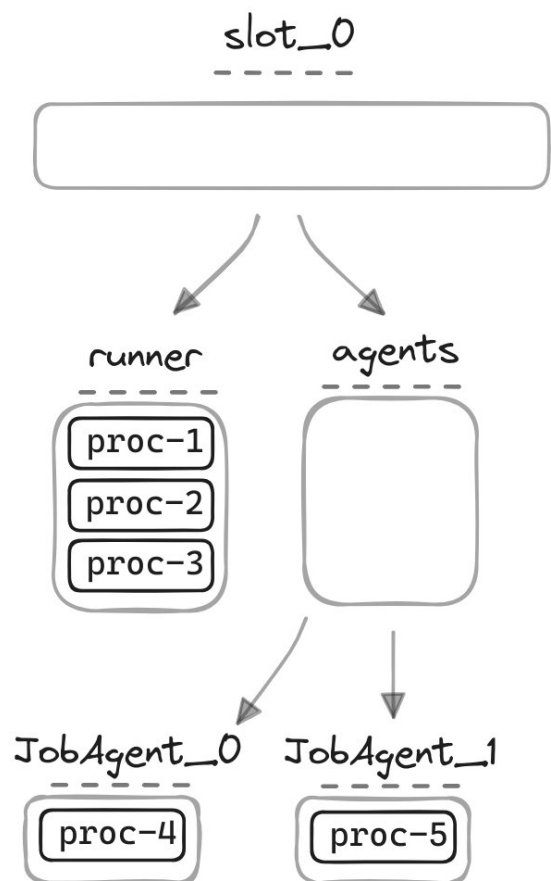
Delagate controllers
(via `cgroup.subtree_control`)



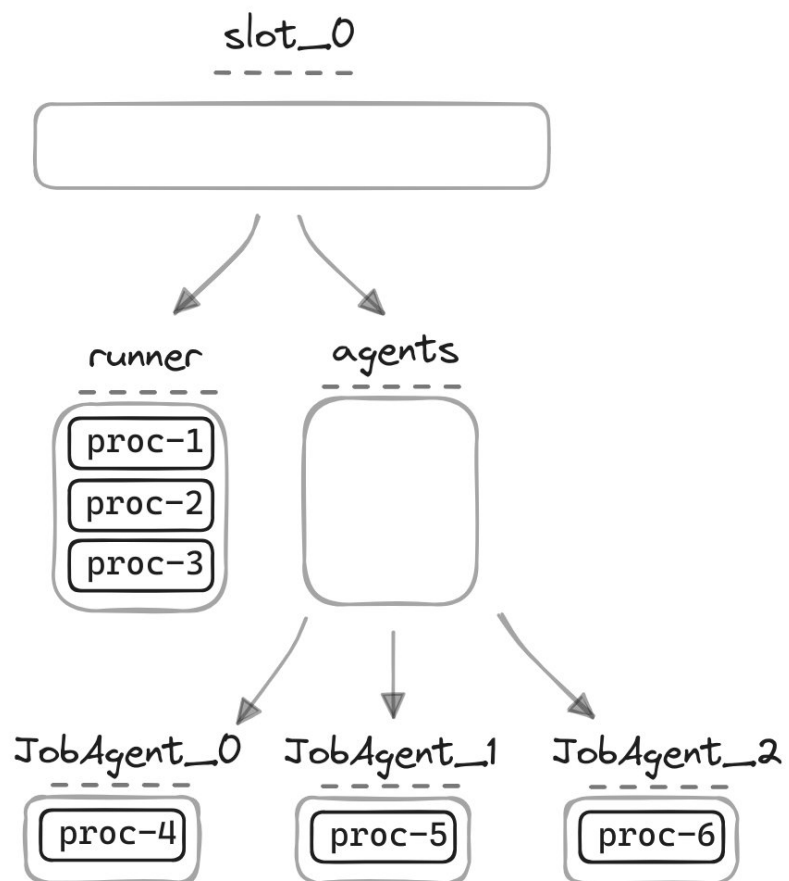
Constructing cgroup in JR/JA



Constructing cgroup in JR/JA



Constructing cgroup in JR/JA



JobRunner/JobAgent cgroup creation

```
├─user
├─tasek_0
├─runner
├─┬-278501 /bin/bash /extra/scratch/tmp/agent.startup.1448512_1699533762880
├─┬-278599 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms16M -Xmx128M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -cp /extra/scratch/alieevee-shared/alien-users.jar alien.site.JobRunner
├─agents
├─┬JobAgent_8
├─┬-280682 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-286991 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975431/tmp/.jalienTimes-2955975431-execution /extra/scratch/workdir/alien-job-2955975431/sampleMemoryConsumer.sh
├─┬-286992 /bin/bash /extra/scratch/workdir/alien-job-2955975431/sampleMemoryConsumer.sh
├─┬-286998 ./a.out
├─┬JobAgent_6
├─┬-280144 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-286862 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975316/tmp/.jalienTimes-2955975316-execution /extra/scratch/workdir/alien-job-2955975316/sampleMemoryConsumer.sh
├─┬-286863 /bin/bash /extra/scratch/workdir/alien-job-2955975316/sampleMemoryConsumer.sh
├─┬-286869 ./a.out
├─┬JobAgent_15
├─┬-282708 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-287752 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975438/tmp/.jalienTimes-2955975438-execution /extra/scratch/workdir/alien-job-2955975438/sampleMemoryConsumer.sh
├─┬-287753 /bin/bash /extra/scratch/workdir/alien-job-2955975438/sampleMemoryConsumer.sh
├─┬-287762 ./a.out
├─┬JobAgent_4
├─┬-279729 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-286607 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975298/tmp/.jalienTimes-2955975298-execution /extra/scratch/workdir/alien-job-2955975298/sampleMemoryConsumer.sh
├─┬-286608 /bin/bash /extra/scratch/workdir/alien-job-2955975298/sampleMemoryConsumer.sh
├─┬-286614 ./a.out
├─┬JobAgent_13
├─┬-281922 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-287281 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975436/tmp/.jalienTimes-2955975436-execution /extra/scratch/workdir/alien-job-2955975436/sampleMemoryConsumer.sh
├─┬-287282 /bin/bash /extra/scratch/workdir/alien-job-2955975436/sampleMemoryConsumer.sh
├─┬-287288 ./a.out
├─┬JobAgent_2
├─┬-279207 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-286339 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975296/tmp/.jalienTimes-2955975296-execution /extra/scratch/workdir/alien-job-2955975296/sampleMemoryConsumer.sh
├─┬-286340 /bin/bash /extra/scratch/workdir/alien-job-2955975296/sampleMemoryConsumer.sh
├─┬-286346 ./a.out
├─┬JobAgent_11
├─┬-281412 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-287151 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975434/tmp/.jalienTimes-2955975434-execution /extra/scratch/workdir/alien-job-2955975434/sampleMemoryConsumer.sh
├─┬-287152 /bin/bash /extra/scratch/workdir/alien-job-2955975434/sampleMemoryConsumer.sh
├─┬-287158 ./a.out
├─┬JobAgent_0
├─┬-278843 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-285495 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975294/tmp/.jalienTimes-2955975294-execution /extra/scratch/workdir/alien-job-2955975294/sampleMemoryConsumer.sh
├─┬-285496 /bin/bash /extra/scratch/workdir/alien-job-2955975294/sampleMemoryConsumer.sh
├─┬-285504 ./a.out
├─┬JobAgent_9
├─┬-280898 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-287054 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975432/tmp/.jalienTimes-2955975432-execution /extra/scratch/workdir/alien-job-2955975432/sampleMemoryConsumer.sh
├─┬-287055 /bin/bash /extra/scratch/workdir/alien-job-2955975432/sampleMemoryConsumer.sh
├─┬-287061 ./a.out
├─┬JobAgent_7
├─┬-280444 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-286919 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975317/tmp/.jalienTimes-2955975317-execution /extra/scratch/workdir/alien-job-2955975317/sampleMemoryConsumer.sh
├─┬-286920 /bin/bash /extra/scratch/workdir/alien-job-2955975317/sampleMemoryConsumer.sh
├─┬-286926 ./a.out
├─┬JobAgent_5
├─┬-279937 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError="echo 'Process %p has run out of memory' > ./29f
├─┬-286660 /usr/bin/time -p -o /extra/scratch/workdir/alien-job-2955975315/tmp/.jalienTimes-2955975315-execution /extra/scratch/workdir/alien-job-2955975315/sampleMemoryConsumer.sh
```



ALICE

JobRunner/JobAgent cgroup creation

```
user
├─task_0
│   └─runner
│       ├──178676 /bin/bash /extra/scratch/tmp/agent.startup.1436588_1699531480866
│       ├──178938 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms16M -Xmx128M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -cp /extra/scratch/alieevee-shared/alien-users.jar alien.site.JobRunner
│       ├──186100 Apptainer runtime parent
│       ├──186165 apptinit
│       ├──186212 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──186616 Apptainer runtime parent
│       ├──186682 apptinit
│       ├──186719 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──187128 Apptainer runtime parent
│       ├──187181 apptinit
│       ├──187233 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──187823 Apptainer runtime parent
│       ├──187888 apptinit
│       ├──187926 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──188443 Apptainer runtime parent
│       ├──188500 apptinit
│       ├──188519 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──189074 Apptainer runtime parent
│       ├──189074 apptinit
│       ├──189102 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──189684 Apptainer runtime parent
│       ├──189738 apptinit
│       ├──189756 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──190303 Apptainer runtime parent
│       ├──190343 apptinit
│       ├──190361 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──190849 Apptainer runtime parent
│       ├──190967 apptinit
│       ├──190985 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──191643 Apptainer runtime parent
│       ├──191663 apptinit
│       ├──191680 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──192268 Apptainer runtime parent
│       ├──192286 apptinit
│       ├──192306 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──192942 Apptainer runtime parent
│       ├──192950 apptinit
│       ├──192976 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│       ├──193606 Apptainer runtime parent
│       ├──193622 apptinit
│       ├──193645 /bin/bash -c source <(/cvmfs/alice.cern.ch/bin/alienv printenv JALIE/1.6.8-1 && echo export APMON_CONFIG=alieevee-wn-1.cern.ch && echo export JOB_CONTAINER_PATH=/cvmfs/alice.cern.ch/containers/fs/singularity/centos7 ); /cvmfs/alice.cern.ch/java
│   └─agents
│       ├──JobAgent_8
│       │   ├──191234 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError=echo 'Process %p has run out of memory' > ./2955963224.oom -Djobagent.vmid=2955963224
│       │   ├──199499 Apptainer runtime parent
│       │   ├──199515 /usr/bin/time -p -o /workdir/tmp.jalientimes-2955963224-execution /workdir/sampleMemoryConsumer.sh
│       │   ├──199540 /bin/bash /workdir/sampleMemoryConsumer.sh
│       │   └──199546 ./a.out
│       ├──JobAgent_6
│       │   ├──189972 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError=echo 'Process %p has run out of memory' > ./2955963222.oom -Djobagent.vmid=2955963222
│       │   ├──198858 Apptainer runtime parent
│       │   ├──198875 /usr/bin/time -p -o /workdir/tmp.jalientimes-2955963222-execution /workdir/sampleMemoryConsumer.sh
│       │   ├──198895 /bin/bash /workdir/sampleMemoryConsumer.sh
│       │   └──198907 ./a.out
│       ├──JobAgent_4
│       │   ├──188720 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError=echo 'Process %p has run out of memory' > ./2955963220.oom -Djobagent.vmid=2955963220
│       │   ├──198117 Apptainer runtime parent
│       │   ├──198131 /usr/bin/time -p -o /workdir/tmp.jalientimes-2955963220-execution /workdir/sampleMemoryConsumer.sh
│       │   ├──198149 /bin/bash /workdir/sampleMemoryConsumer.sh
│       │   └──198155 ./a.out
│       ├──JobAgent_2
│       │   ├──187426 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError=echo 'Process %p has run out of memory' > ./2955963218.oom -Djobagent.vmid=2955963218
│       │   ├──197453 Apptainer runtime parent
│       │   ├──197470 /usr/bin/time -p -o /workdir/tmp.jalientimes-2955963218-execution /workdir/sampleMemoryConsumer.sh
│       │   ├──197489 /bin/bash /workdir/sampleMemoryConsumer.sh
│       │   └──197495 ./a.out
│       └──JobAgent_11
│           ├──193264 /cvmfs/alice.cern.ch/java/JDKs/x86_64/jdk-latest/bin/java -client -Xms50M -Xmx50M -Djdk.lang.Process.launchMechanism=vfork -XX:+UseSerialGC -XX:OnOutOfMemoryError=echo 'Process %p has run out of memory' > ./2955963228.oom -Djobagent.vmid=2955963228
│           ├──200333 Apptainer runtime parent
│           ├──200346 /usr/bin/time -p -o /workdir/tmp.jalientimes-2955963228-execution /workdir/sampleMemoryConsumer.sh
│           ├──200366 /bin/bash /workdir/sampleMemoryConsumer.sh
│           └──200373 ./a.out
```



Site requirements for cgroups v2

Site requirements for cgroups v2

- Must be an OS with support for Cgroups v2, e.g.
 - **EL 9** (Recommended)
 - **EL 8*** (Needs extra workaround if on Slurm)
- Must be **enabled** (default on EL9)
- **HTCondor 23.2** (i.e. not LTS)
- Or, **Slurm 22.05+**
 - For now, still with custom plugin for setup
 - Alternatively, can also be done via Slurm *prolog/epilog scripts*
 - Additional **workaround needed** for **EL 8**
 - Kernel missing “*cgroups.kill*” feature, used for cleanup
 - If cleanup fails, WN will go into drain state

Site requirements for cgroups v2

- Must be an OS with support for Cgroups v2, e.g.
 - **EL 9** (Recommended)
 - **EL 8*** (Needs extra workaround if on Slurm)
- Must be **enabled** (default on EL9)
- **HTCondor 23.2** (i.e. not LTS)
- Or, **Slurm 22.05+**
 - For now, still with custom plugin for setup
 - Alternatively, can also be done via Slurm *prolog/epilog scripts*
 - Additional **workaround needed** for **EL 8**
 - Kernel missing “*cgroups.kill*” feature, used for cleanup
 - If cleanup fails, WN will go into drain state

In other words, if already on **HTCondor 23.2** and **EL 9**, no action needed!

Summary and outlook

- JAliEn **job pilot** is becoming more **flexible**
 - Agnostic setup
 - Automatic matching of
 - Platform/containers
 - Packages
 - Architectures
 - ... including **aarch64**
- New isolation features, such as layered containers and **cgroups v2**
 - ... but needs changes to how CE/LRMS construct/clean cgroups
 - **HTCondor** – all set from **v23.2**
 - **SLURM** – TBD
 - Custom plugin or prolog needed for now
- Nevertheless, functionality **in place** for when support becomes available!



Thank You
[Questions, comments]?
email: mstoretv@cern.ch