Singularity@ATLAS

Andrej Filipcic

Benefits for ATLAS and sites

- Sites can use any host OS of their preference
  - Minimal OS (eg CoreOS), bleeding edge (Fedora), latest enterprise OS
  - Site OS major upgrades not affecting ATLAS production
- Many images can be used simultaneously on the same site, eg
  - SL6 for rel 21 production
  - SL7 for rel 22 testing and validation
  - SL5,4,3 for analysis of old data, data preservation
- Common approach for execution, software distribution for all sites, including HPCs and ATLAS@Home
- Simplification of site requirements
  - Middleware can be provided in the (common) singularity images
  - Needs a discussion within WLCG and wider (including non-LHC VOs)

Deployment

- Large scale singularity deployment, starting with all modern OS sites
  - SL7 and equivalent
  - Debian, ubuntu, SUSE, ... if recent kernel (support for overlay FS and shared mounts)
- Some sites already using it:
  - SiGNET, ARNES, HPC2N, LRZ, MWT2, ...
- Experience:
  - No issues, very robust in the last few months
  - Native performance (as compared to VM)
- Singularity installation required on all centos7 (or similar) sites
- Support for older sites (eg SLES11, or SLC6)
  - NOT to be enabled by default, meant only for sites that want to use it
  - Img needs to have bind mount directories created in advance - they can be requested on per site basis and provided in the common image

ATLAS CVMFS Image(s)

- /cvmfs/atlas.cern.ch/repo/images/
  - docker singularity
- /cvmfs/atlas.cern.ch/repo/images/singularity/def/
  - atlas-centos6.def atlas-centos7.def
- /cvmfs/atlas.cern.ch/repo/images/singularity/
  - x86_64-centos7-20170427.img x86_64-centos6-20170424.img
  - Symlinks: <cmtconfig>.img, eg
    - x86_64-slc6-gcc62-opt.img -> x86_64-slc6.img
    - x86_64-slc6.img -> x86_64-centos6-20170427.img
- For now, images only
- Soon, img unpacked into chroot tree, keeping img and dir in sync
  - Better for cvmfs caching and container startup

Running the jobs

- Full job in container:
  - Some sites using singularity by default -> batch job executed in container automatically
  - singularity: SiGNET with gentoo, HPC2N with debian, LRZ with SLES11...
  - docker: RAL with centos7 starting SLC6 containers
  - Not flexible enough for ATLAS
- Full job after the payload is known:
  - cmtconfig to img mapping
  - pilot starts singularity, not tested yet
- Container per job step (testing, using the same img for all)
  - Stagin
  - Payload (transform) execution
  - Stageout

Open points

- Custom images:
  - ATLAS specific - payload execution - no middleware is needed
  - Generic - stageing, stageout - contains middleware, a good candidate to be shared/supported/validated by WLCG or wider
- Image distribution: cvmfs seems the best option for singularity
  - for docker no decision yet, need to support user containers
- Singularity on docker sites
  - Possible to run singularity within docker
  - To be tested at RAL, but CC7 docker containers need to be enabled (SLC6 only for now)
- Img validation
  - Some sites want to fetch the img and validate (checksum) before enabling

AGIS

- All the site specific settings will be stored in AGIS
- Start with catchall parameter, eg
  - catchall="singularity_bindmounts=/data0,/var/spool/slurmd,/ceph/grid"
- Later on, proper AGIS entries will be defined for full flexibility, eg
  - Supported images
  - Default image for jobs not specifying the container
- Should be enough for pilot to execute the job inside the singularity container

Further steps

- Large scale testing with in-pilot containers
- Review sw requirements for job steps
  - Minimize the sw dependency on parts executing on host OS
  - Probably needs further pilot development and cleanup
- Enable flexible containerization approach in pilot (steps)
- Evaluate different approaches and review site specific requirements to find the best and most common option
- Review the AGIS settings
- Automated img creation (security) and versioning