Containers for ATLAS

Andrej Filipcic

Motivations to use containers

- Similar concept to virtual machines, but much more flexible and “zero” performance loss
  - chroot done properly
- Independence of execution environment vs host OS
- Can provide custom development, testing environment on grid
- Much easier and flexible to use
- Can make site maintenance and central image/software management much more transparent

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
  - OS upgrades can be done on the fly
- 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
- Sites can only provide basic OS on the nodes
- Much more flexible and secure from site point of view
  - Isolation and traceability - replacing glexec functionality
- Common approach for execution, software distribution for all sites, including HPCs and ATLAS@Home

Singularity and Docker

- Many containers technologies (shifter, LXC, LXD...). Focus on
  - Singularity - for production
  - Docker - for development, custom workflow, mainly requested by ATLAS software development
- Docker - more difficult to deploy, usage foreseen on few proactive sites
  - Docker daemon on the nodes, authorized users
  - Root privileges in the container - might not be suitable for large scale deployments
- Singularity - tuned for batch systems (HPCs as well)
  - Simple to install - one rpm, straightforward configuration
  - Non-privileged - UID is kept while starting the container

Deployment

- Already decided (last ATLAS S&C Workshop) for 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)
  - One image can serve them all
- Support for older sites (eg SLES11, or SLC6)
  - Img needs to have bind mount directories created in advance - they can be requested on per site basis and provided in the common image

CVMFS Image(s)

- Bootstrap:
  - Singularity or docker def file, start with singularity for simplicity
  - rpm installation only, no configuration
- Singularity can use local img file, remote (http) files, directory (eg. /cvmfs/cernvm-prod.cern.ch/cvm3)
  - Start with img files in cvmfs (~2GB)
    - May be a problem with sites using very small cvmfs caches, especially if we plan to use several images in the future: we'll have to enforce our requirements
  - Image repository in /cvmfs/atlas.cern.ch/repo/images/singularity/...
  - Multiple versions of the same images? Probably not needed
- How many images?
  - Default (fat) SLC6, SL7 - including grid middleware, development tools
    - Need to understand if we need some autoconfiguration for grid mw tools
  - Start with SLC6
  - We'll need to rebuild the images whenever a security patch is available

Bind mounts

- Two types - default and site specific
- Default:
  - /cvmfs
  - /sys/fs/cgroup
  - /etc/grid-security/certificates
- Site specific:
  - Local scratch
  - Shared FS
- Other singularity options:
  - Isolation: --ipc, --pid, ...

AGIS settings

- 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

pilot

- If singularity is in catchall, the pilot executes the payload by default in the SLC6 container
- Should call singularity as soon as possible, eg right after getting the payload description
  - To support sites with minimal host OS (eg CoreOS)
  - Many sites expressed the wish to have basic, minimal host OS, everything else (grid MW) can be deployed in container images
- Final goal: PanDA should set the container in the job description for every payload
  - Flexibility to execute the payload in any OS (eg, SLC5, SLC6, SL7)
  - Eventually support for non-Intel platforms...

ATLAS longer-term perspective

- ALL the ATLAS jobs will use containers
  - There might be exceptions with specific sites (eg. some HPCs)
- The basic OS on host should be enough - the libs, MW should be provided in containers
  - Easier for sites
  - Centralized SW deployment
- cvmfs will be used as the main distribution point for container images
- Isolation will be used

Points of wider interest

- Image management:
  - Bootstrapping (for both singularity and docker) and deployment - how to manage?
  - Official (signed?) images, private images, approved images?
- Common images:
  - Simplified middleware deployment
  - Staged bootstrapping - eg common core + VO specific part
- Security:
  - Procedure to deal with security vulnerabilities
  - Tracing the container activity - clear instructions for sysadmins
  - Guidelines for site deployment, in particular for docker (eg avoid access to shared FS inside the container)

Implications for WLCG

- Minimal host OS - not compatible with WLCG site requirements
  - Eg, middleware is missing, needs to be provided through the container
- Need to agree on default deployment model for VOs not using the containers, singularity
  - Default container
  - image/directory location and maintenance (can it be cvm3?)
  - How to integrate it in CEs?
- Check with others, (eg Belle2 Dirac) for preferred deployment model and execution strategy
- Agree on recommendations:
  - Traceability and isolation

Conclusions

- ATLAS is moving towards full containerization for production and analysis
  - Simplifies site maintenance and centralizes deployment
- Several details to be addressed, many are ATLAS specific
- WLCG needs to decide whether to keep containers at VO level, or rather fully embrace it and adapt the distributing computing model to work transparently
- Many sites are proactive and interested in containers, some wish to have minimal host OS
- Time for a Task Force?