Singularity@SiGNET

Andrej Filipčič

SiGNET

- 4.5k cores, 3PB storage, 4.8.17 kernel on WNs and Gentoo host OS
- 2 ARC-CEs with 700TB cephfs ARC cache and 3 data delivery nodes for input/output file staging
- Jobs ran in payload push mode

- Some history
  - Opterons in 2003, SLC3 did not work on that hardware
  - Installed custom OS (gentoo) -> jobs executed in chroot with SLC tree
  - Difficulties - many tweaks:
    - Autofs
    - “Manual” bind mounts, relatively complex setup with schroot

Singularity approach

- Easier and more intelligent chroot
- How to enable:
  - Install singularity pkg, minor configuration changes
  - Prepare SLC6 image with HEP OSlibs
  - Hack batch submission script to call singularity

Singularity installation and configuration

- Install rpm, deb or ebuild or whatever, a single package
- Configure local behaviour
- That's it:
  - No daemon
  - Any user can run containers
- Caveat:
  - SLC6 works but some issues with overlay and autofs
- All modern OSes supported

No other actions needed on the sites/nodes!

```
/etc/singularity/singularity.conf
# Should we allow users to request the PID namespace?
allow pid ns = yes
...
# only newer kernels, RH7.3 is OK
enable overlay = yes
...
mount home = yes
...
#bind path = /opt
#bind path = /scratch
bind path = /etc/hosts
user bind control = yes
...
# for autofs, cvmfs
mount slave = yes
```

Preparing the container image

singularity create atlas.img

singularity bootstrap atlas.img atlas.def

Container is ready to use

No setup needed inside the container

- Users, groups on demand
- Config (/etc/...) through bind mounts

Atlas.def (a bit simplified)

```
----------
BootStrap:yum
OSVersion: 6.8
MirrorURL: http://ftp.arnes.si/mirrors/centos.org/6.8/os/x86_64/
UpdateURL: http://ftp.arnes.si/mirrors/centos.org/6.8/os/x86_64/
Include: yum git

#UpdateURL: http://mirror.centos.org/centos-%{OSVERSION}/%{OSVERSION}/updates/$basearch/

%setup

%runscript
    echo "Running the container..."

%post
    echo "Installing the packages inside the container"
     rpm --rebuilddb
     yum -y install vim-minimal
    echo "Installing Development tools"
     yum -y groupinstall "Development Tools"
     yum -y install wget
    wget http://linuxsoft.cern.ch/wlcg/sl6/x86_64/HEP_OSlibs_SL6-1.0.16-0.el6.x86_64.rpm
    yum -y localinstall HEP_OSlibs_SL6-1.0.16-0.el6.x86_64.rpm
    yum -y update
```

Using the container

```
prdatl01@f9nd007 ~ $ more /etc/lsb-release
DISTRIB_ID="Gentoo"
prdatl01@f9nd007 ~ $ singularity shell /ceph/sys/singularity/container/atlas.img
Singularity: Invoking an interactive shell within container...

Singularity.atlas.img> more /etc/redhat-release
CentOS release 6.8 (Final)
Singularity.atlas.img> exit

prdatl01@f9nd007 ~ $ singularity exec /ceph/sys/singularity/container/atlas.img /bin/pwd
/ceph/grid/home/prdatl01

prdatl01@f9nd007 ~ $ id
uid=11951(prdatl01) gid=11950(prdatlas) groups=11950(prdatlas),11000(lcgatlas)
prdatl01@f9nd007 ~ $ singularity exec /ceph/sys/singularity/container/atlas.img id
uid=11951(prdatl01) gid=11950(prdatlas) groups=11950(prdatlas),11000(lcgatlas)
```

Executing the jobs in containers (SLURM)

SLURM script follows:

```
------------------------------------------------------------------
#!/bin/bash -l
# SLURM batch job script built by grid-manager
#SBATCH --no-requeue
#SBATCH --mem_bind=verbose,local
#SBATCH -e /ceph/grid/session/9oQODmGTV5pn4J8tmqCBXHLnABFKDmABFKDmEKFKDmABFKDm7SMDzm.comment
#SBATCH -o /ceph/grid/session/9oQODmGTV5pn4J8tmqCBXHLnABFKDmABFKDmEKFKDmABFKDm7SMDzm.comment

#SBATCH -p batch
#SBATCH --nice=11
#SBATCH -J 'user_khill_002_'
#SBATCH --get-user-env=10L
#SBATCH -n 1
#SBATCH
#SBATCH -t 5760:0
#SBATCH -t 5760:0
#SBATCH --mem-per-cpu=2000
if [ -z $SINGULARITY_CONTAINER ]; then
  exec /usr/bin/singularity exec -B
/etc/grid-security/certificates,/var/spool/slurmd,/cvmfs.local:/cvmfs,/ceph/grid,/data0,/sys/fs/cgroup
/ceph/sys/singularity/container/atlas.img $0
fi
# Rest of the script...
```

Note on batch jobs

- The bind mounts need to be explicitly specified
  - cvmfs, local workdir, shared storage, cgroups, /etc/grid-security/certificates ...
  - They could be all specified in the system config file as well, but then less flexible
- If overlay is not used, all paths for bind mounts must be present in the image
  - not practical for generic images for all the sites
- Mount points such as /net or /cvmfs (autofs) needs to be shared-mounted to work transparently
  - mount --make-rshared /cvmfs
  - But /net does not work with old kernels (eg SLC6 host OS)

Running ATLAS job in singularity (on ARC-CE)

- Modify submit-SLURM-job script to insert the singularity exec line at the top of the generated SLURM script (just as an example, re-executes the whole script inside the container)

```
...
if [ ! -z "$joboption_memory" ] ; then
  echo "#SBATCH --mem-per-cpu=${joboption_memory}" >> $LRMS_JOB_SCRIPT
fii
. ${pkgdatadir}/choose-singularity.sh >>  $LRMS_JOB_SCRIPT
....
if [ ! -z "${joboption_runtime_0}" ] ; then
  if [[ $joboption_runtime_0 =~ APPS/HEP/ATLAS(\.*) ]] ; then
    echo 'if [ -z $SINGULARITY_CONTAINER ]; then'
    echo '  exec /usr/bin/singularity exec -B /etc/grid-security/certificates,/var/spool/slurmd,/cvmfs.local:/cvmfs,/ceph/grid,/data0,/sys/fs/cgroup
/ceph/sys/singularity/container/atlas.img $0'
    echo 'fi'
    #clean env
    echo 'unset PYTHIA8'
    echo 'unset PYTHIA8DATA'
  fi
fi
```

Other ways to run

- Batch wrapper script executes the pilot job in container
  - singularity exec … python pilot.py -h SiGNET -s SiGNET
- The pilot script executes the given payload in container
  - singularity exec …. python RunJob.py …
  - Container can be fixed by PanDA on per job level
- Transform is executed in container
  - singularity exec … python athena.py …

- The last one is most flexible:
  - Multi-step jobs can use their own containers
  - Multi-pilot jobs….

ATLAS production with singularity

- Switched SiGNET cluster for all ATLAS jobs 3 weeks ago
  - Then switched for all jobs (eg nagios probes, Belle-2, local...) 2 weeks ago
- Switched ARNES clusters (4.4k cores) for ATLAS jobs to singularity 1 week ago
  - SLC6 host OS, SLC6 images - proof of concept
  - Cluster will be soon migrated to CC7 but will continue to run ATLAS jobs in SLC6 containers
- No major issues
  - Added few missing rpms in the image
  - Move HOME for grid jobs from autofs to statically mounted dir on ARNES (due to SLC6)

Benefits

- Custom universal OS images - all required software is available everywhere and simple to deploy, extend
  - Eg client middleware
  - Each experiment can use their own image and manage its own software
- Decoupling host OS from executing OS
  - Job can run inside any OS on all sites
  - Upgrading the sites decoupled from experiment OS requirements
- Minimal performance impact (check LBL slides linked in indico)

What about docker

- A bit more complex to setup
  - Docker daemon needs to run
  - User is root inside the container
- BUT: docker images can be executed with singularity
  - Either directly:
    - singularity shell docker://ubuntu:latest
  - Or by transforming the image:
    - singularity create ubuntu.img
    - singularity import ubuntu.img docker://ubuntu:latest

Docker could be used to manage software, but singularity is more convenient for execution

What does a site need to run ATLAS jobs?

- Singularity package
- /cvmfs on host OS
- Updated certificates
- Everything else can be provided inside the singularity image !

- For HPCs:
  - singularity enabled nodes should be enough
  - System and ATLAS software can be provided by the image