

# **Singularity containers & legacy code @ CentOS7**

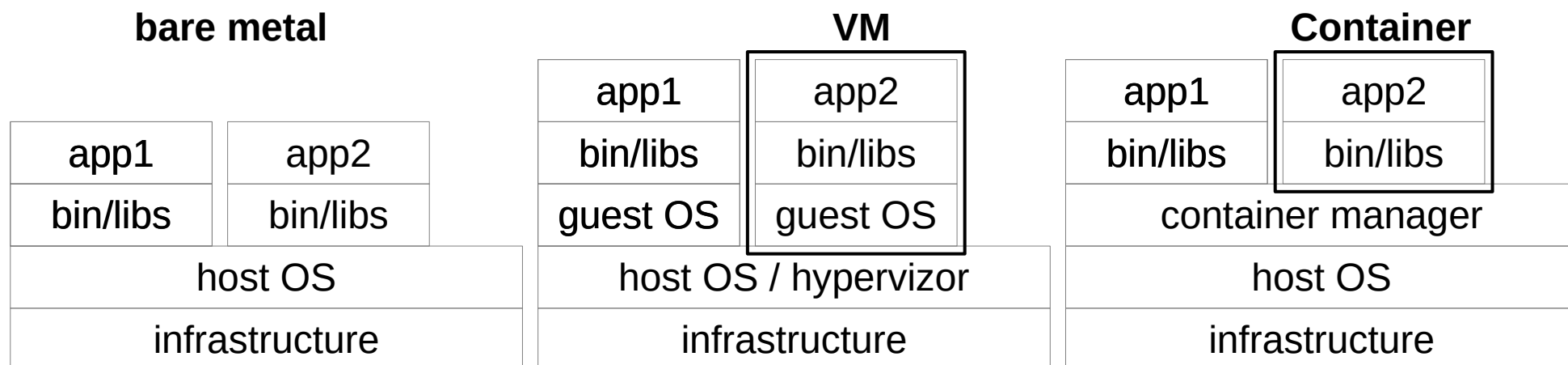
Petr Vokáč

FZU Farm users meeting 2019  
24<sup>th</sup> March

# Motivation

- SLC6 EOL at the end of 2020 → CentOS7
- FZU Farm services & grid middleware already upgraded C7
- Difficult to coordinate upgrade with all experiments / users
  - WLCG use C7 by default since beginning of April (ui & batch)
  - ATLAS would like to have most resources on C7 till summer 2019
  - Alice jobs can still run on both SLC6 & C7
  - Auger jobs use SLC6 (can run on C7)
  - NOvA jobs require SLC6
  - DUNE jobs use SLC6 and SLC7
  - local users
- Code compiled with C7 doesn't work with older OS / libraries
- Better support for new hardware and additional OS features
- Compiler (standard, performance, hardware, ...)

# Containers



- Address operational issues for cluster worker nodes
  - different OS requirements
    - upgrades no longer needs coordination between experiments
    - conflicting library version requirements
  - minimal OS installation + container support
    - e.g. HEP\_OSlibs, EGI vs. OSG CA trust anchors, UMD, OSG, WLCG
  - allows users / experiments to completely define their environment
  - payload isolation / data + software preservation / reproducible
  - user containers

# Singularity

- Linux container management developed for HPC
  - requires recent OS features available only in CentOS7.6
  - easy deployment – no running daemon (“one” package + simple cfg)
  - no special privileges necessary to be used securely by users
  - doesn’t need special support from local batch system
    - htcondor supports singularity, docker, vm, ...
- Image based software distribution
  - supports own image format and also docker images
  - filesystem based images can be provided directly from CVMFS
- Documentation
  - <https://singularity.lbl.gov/> → <https://www.sylabs.io/docs/>
- Installed on CentOS7 worker nodes and user interface
  - currently version 3.1.x
  - very active development (generally for all container technologies)

# Using singularity

- Containers supported by [experiment](#) (sl5, sl6, sl7, slc6, centos7, ...)

```
[user@ui1 ~]$ setupATLAS -c slc6
```

- Container images available in CVMFS (e.g. FZU SLC6 WN)

```
[user@ui1 ~]$ cat /etc/redhat-release  
CentOS Linux release 7.6.1810 (Core)  
[user@ui1 ~]$ singularity exec \  
/cvmfs/farm.particle.cz/singularity/fzu_wn-slc6-latest cat /etc/redhat-release  
Scientific Linux release 6.10 (Carbon)
```

- Images available in Singularity / Docker hub (don't use in batch)

```
[user@ui1 ~]$ singularity search ubuntu  
...  
library://library/default/ubuntu  
Tags: 14.04 16.04 18.04 18.10 latest  
...  
[user@ui1 ~]$ singularity exec \  
library://library/default/ubuntu:18.10 cat /etc/os-release | head -n 2  
NAME="Ubuntu"  
VERSION="18.10 (Cosmic Cuttlefish)"
```

# Using singularity

- CLI provides basic help with examples, details in [documentation](#)

```
[user@ui1 ~]$ singularity help
Usage:
  singularity [global options...] <command>
...
[user@ui1 ~]$ singularity help exec
...
Examples:
  $ singularity exec ...
...
```

- Singularity interactive shell with current FZU Farm SLC6 OS

```
[user@ui1 ~]$ singularity shell /cvmfs/farm.particle.cz/singularity/fzu_wn-slc6-latest
sh-4.1$ cat /etc/redhat-release; id
Scientific Linux release 6.10 (Carbon)
uid=8021(vokac) gid=100(users) groups=100(users),1700(fermilab)
```

- Some experiments provide own general purpose images
  - /cvmfs/atlas.cern.ch/repo/containers/fs/singularity
  - /cvmfs/unpacked.cern.ch/registry.hub.docker.com/library/
  - /cvmfs/singularity.opensciencegrid.org/opensciencegrid

# Using singularity

- Isolation – namespaces (pid, mount, user, ...)
  - limited list of processes
  - different filesystem view (e.g. /tmp)
    - root filesystem from singularity image
    - bind mounted some required locations
      - /etc/localtime
      - /etc/hosts
      - /etc/grid-security/certificates
      - /cvmfs
      - /mnt
    - home directory also automatically bind mounted
    - user allowed to bind mount their own locations

```
[user@ui1 ~]$ singularity shell -B /mnt/nfs17:/data17 \  
/cvmfs/farm.particle.cz/singularity/fzu_wn-slc6-latest  
sh-4.1$ ls /data17  
ATLASDiffXsecAllHadBoosted bakalova blazekj chudoba ...
```

- Running application directly from singularity template with “run”

# Singularity @ local batch

- Grid / pilot support for singularity
  - doesn't need any input from user for production
  - pilot automatically detects and runs payload in right container
  - user analysis jobs allow to specify containers (e.g. ATLAS prun)
- HTCondor batch container support
  - docker containers
    - not supported @ FZU
    - can't easily support user containers / images
  - singularity available on all CentOS7 WN and UI
  - job executed directly in requested SingularityImage
- Wrapper / singularity executed directly from script
  - run every command in different containers
  - latest singularity supports container in container
    - you can use batch containers for “pilot” and different for analysis code



# Singularity @ local batch

- Same htcondor universe
  - no modifications to existing jdl
  - vanilla is default (can be omitted)
- Additional requirements
  - HasSingularity (can be omitted)

```
[user@ui1 ~]$ condor_status \  
-af Name HasSingularity
```

- **SingularityImage** – location / url of required image
- custom htcondor attribute TestCentOS7
  - this requirements will be dropped (today?)
  - only necessary during SLC6 → C7 transition for non-containerized jobs

```
universe = vanilla  
executable = job.sh  
  
output = job.o$(ClusterId)  
error = job.e$(ClusterId)  
log = job.l$(ClusterId)  
  
Requirements = \  
                HasSingularity  
#+SingularityBind = ...  
+SingularityImage = \  
    /cvmfs/atlas.cern.ch/repo\  
    /containers/fs\  
    /singularity/x86_64-slc6  
  
+TestCentOS7 = True  
+MaxRuntime = 15*60  
  
queue
```

# BACKUP

# Submit file

- Requirements
  - constrain machines suitable for our jobs
  - standard options
    - request\_\*
  - user defined
    - expression matching for job / machine classad

```
executable = job.sh
arguments = arg1 arg2 arg3

request_cpus = 2
request_memory = 2GB
request_disk = 10GB
requirements = has_sse4_1 && HasVM

queue
```

```
[user@ui ~]$ condor_q -af:r Requirements
(TARGET.Arch == "X86_64") && (TARGET.OpSys == "LINUX") \
&& (TARGET.Disk >= RequestDisk) && (TARGET.Memory >= RequestMemory) \
&& ((TARGET.HasFileTransfer) || (TARGET.FileSystemDomain == MY.FileSystemDomain))
```

# Submit file

```
[user@ui ~]$ condor_status -compact
Machine Platform Slots Cpus Gpus TotalGb FreCpu FreeGb
aplex01.farm.particle.cz x64/SL6 31 32 76.17 1 64.30
aplex02.farm.particle.cz x64/SL6 27 32 76.17 1 18.80
...
rubus23.farm.particle.cz x64/SL6 25 32 76.17 0 11.42
wntestcc7.farm.particle.cz x64/CentOS7 0 2 5.86 2 5.86
```

```

Machines Owner Claimed Unclaimed Matched Preempting Drain
x64/CentOS7 1 1 0 0 0 0 0
x64/SL6 4443 11 4389 1 0 0 42
Total 4444 12 4389 1 0 0 42
```

```
[user@ui ~]$ condor_status -startd -long
```

```
...
[user@ui ~]$ condor_status -startd \
  -constraints '(Machine == "db1.farm.particle.cz")||(Machine ==
  "db2.farm.particle.cz")'
```

```
...
[user@ui ~]$ condor_status -startd \
  -constraints 'stringListMember(Machine,
  "db1.farm.particle.cz,db2.farm.particle.cz")'
```

```
...
[user@ui ~]$ condor_status -startd \
  -constraints 'regexp("^db[0-9]+\.\farm\.particle\.cz$", Machine, "I")'
```

# Docker

- not supported by batch @ FZU
- docker images can be used with singularity

```
universe = docker  
executable = job.sh  
arguments = arg1 arg2 arg3  
  
output = job.o$(ClusterId)  
error = job.e$(ClusterId)  
log = job.l$(ClusterId)  
  
#+OS_IMAGE = SLC5  
+OS_IMAGE = SLC6  
#+OS_IMAGE = CC7  
#+OS_IMAGE = Ubuntu  
  
+MaxRuntime = 15*60  
  
queue
```