Singularity at The RACF/SDCC

Chris Hollowell <hollowec@bnl.gov>, Xin Zhao <xzhao@bnl.gov>

HEPiX Fall 2017
KEK - Tsukuba, Japan

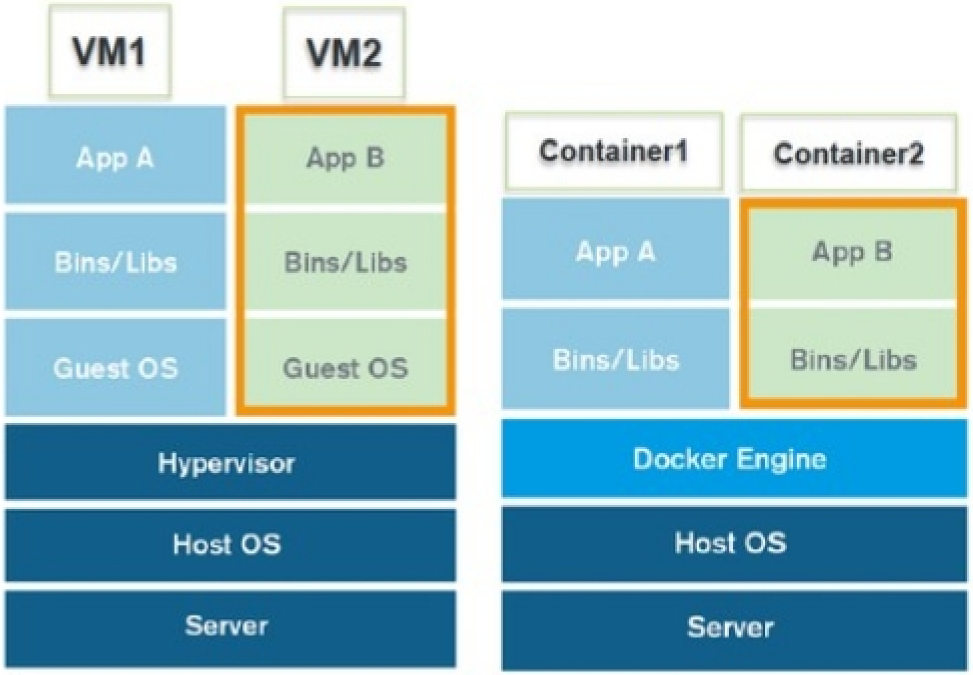
70 YEARS OF DISCOVERY
A CENTURY OF SERVICE

U.S. DEPARTMENT OF ENERGY

BROOKHAVEN NATIONAL LABORATORY

- What is Singularity?
  - OS container management system for Linux developed at LBNL
    - Containers are an OS abstraction providing enhanced process isolation
      - Depending on the level of isolation required, processes in containers can appear to be running in completely different OS instances
      - Processes in distinct Singularity containers can exist with different Linux userlands, but are all running under the same OS kernel
        - Strict Linux kernel syscall ABI backwards compatibility and relative stability makes this possible
          - For instance, one can run SL6 and Debian 7 containers under SL7 without issue
      - Primary difference between “containers” and “virtualization” is that with containers the full hardware stack is **not** abstracted: only the OS is
        - Cannot run a completely different OS in a container: i.e., you cannot start a Windows container on a Linux host

- What is Singularity (Cont.)?



Layers of Hardware vs OS Virtualization

- Similar to Docker, LXC, etc.
  - Makes use of Linux kernel “namespaces” to implement containerizaton
- Different in that it was specifically designed to enable mobility of compute for HPC/HTC jobs

- Reasons to Use Singularity in HTC/HPC Jobs
  - Divorces a job's software requirements from what's actually available in the host OS. Administrators can (almost) run whatever flavor and version of Linux they'd like on their physical nodes, as long as users create containers with all of their software dependencies inside
    - Essentially gives users freedom to run their jobs wherever there's a Linux host with Singularity installed
  - Administratively straightforward and lightweight to mange: install a single package (and potentially make minor configuration changes), and you're done.
    - Unlike Docker, there is no daemon to start, and no need to add users to a special group in order to execute jobs in containers
    - Doesn't make use of namespaces which aren't typically required by compute jobs
      - By default, PID and network namespaces are the same as the host OS'
    - Secure out of the box - regular users are never root in a container
      - setuid permissions not required with newer distros (RHEL 7.4+)
    - Built-in support in new versions of HTCondor (8.6.0+)

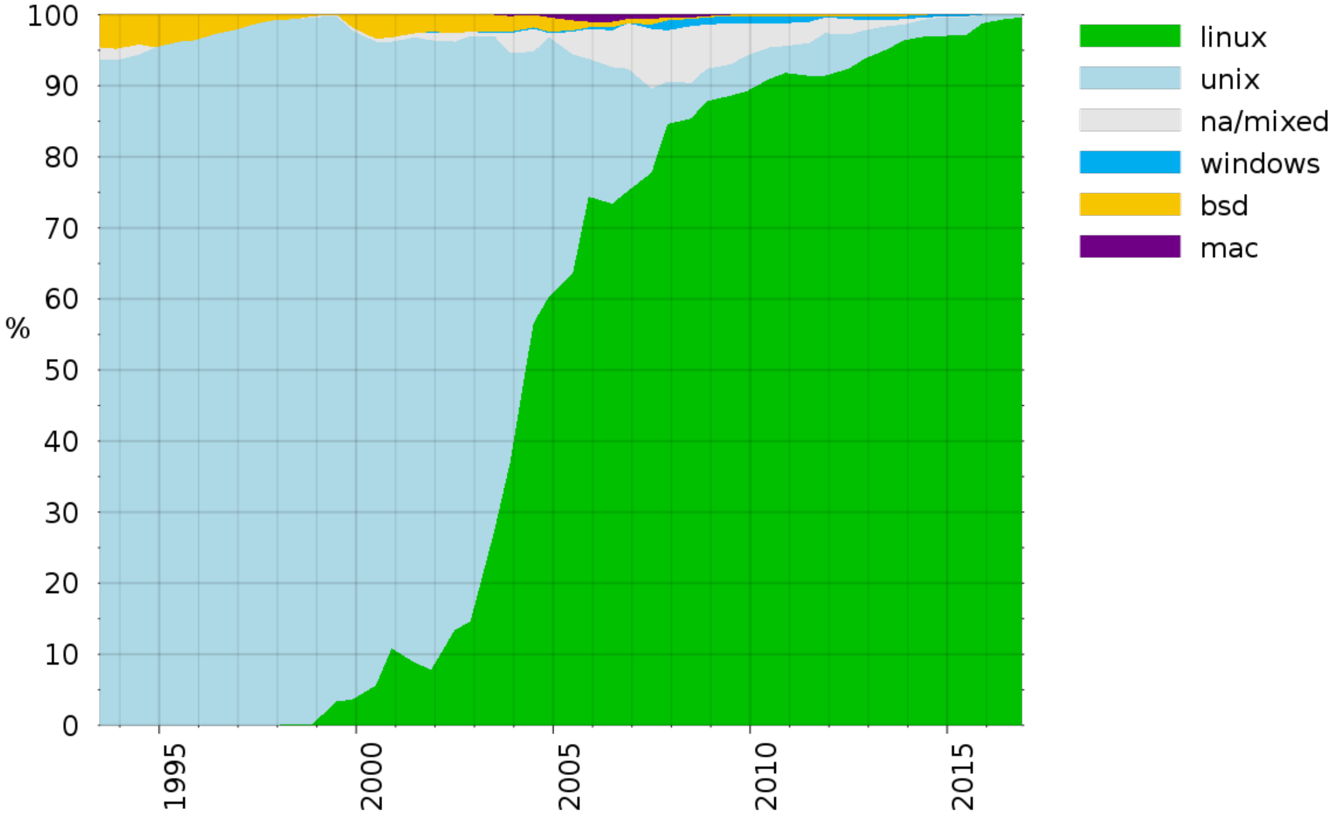
Containers vs VMs For HTC/HPC Processing Workloads

- No time wasted booting or shutting down VMs
  - Singularity containers can be instantiated in milliseconds
- No abstraction/emulation of hardware means no performance loss or additional latencies associated with these abstractions
  - Processes in containers can still have direct access to GPUs, IB networks
  - Direct access to filesystems mounted in the host OS can be made available, if desired. No reduction in performance from implementing virtual block devices
  - CPU performance is somewhat reduced by virtualization
    - In 2015, CERN investigated KVM optimization in order to reduce CPU performance degradation associated with virtualizing workernodes – up to a ~20% HS06 reduction was noted for VMs, compared to bare metal, before their optimizations
      - Even after optimization, CPU (HS06) performance was still reduced by a few percent for VMs vs bare metal

Containers vs VMs For HTC/HPC Processing Workloads (Cont.)

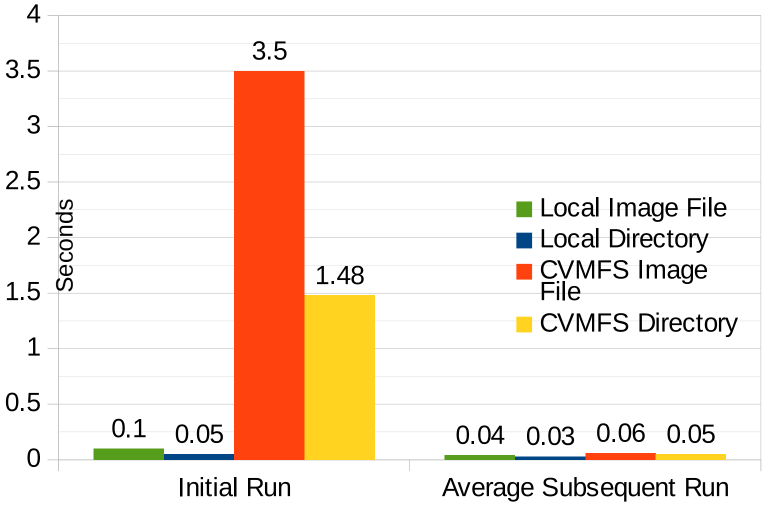
- Full details from HEPiX 2015: https://indico.cern.ch/event/384358/contributions/909247/
- Studies by myself and others have shown there to be no HS06 CPU performance reduction when running in Linux containers
  - RACF study from 2015 (was performed with Docker, but using the same underlying Linux kernel namespace container technology as Singularity):
    - https://indico.cern.ch/event/384358/contributions/909242/
  - Verified no HS06 loss under Singularity
- Over the years, scientific computing has converged toward using Linux as the operating system of choice for batch processing
  - As of Nov 2016, 99.6% of the supercomputers in the TOP500 list ran Linux
  - As a result, hypervisor-level hardware virtualization support is largely unnecessary on HPC/HTC nodes

- Containers vs VMs For HTC/HPC Processing Workloads (Cont.)



OS share on TOP500 Supercomputers since 1993 (source: Wikipedia)

- Singularity Container Formats
  - Supports:
    - Image files (ext3/squashfs)
    - Tarballs
    - Directories (AKA sandbox, or unpacked)
  - Can directly run Docker images
  - Is there a performance difference when using image files vs directories?
    - Directory-based distribution of containers is the most efficient for network filesystems with local caches: i.e. CVMFS, AFS, etc.
      - Distributing many large single/complete image files may stress caches
      - ATLAS recently released directory-based distributions of their Singularity containers in CVMFS
        - /cvmfs/atlas.cern.ch/repo/containers/fs/singularity
    - Other than that, only very minimal performance differences encountered in our testing

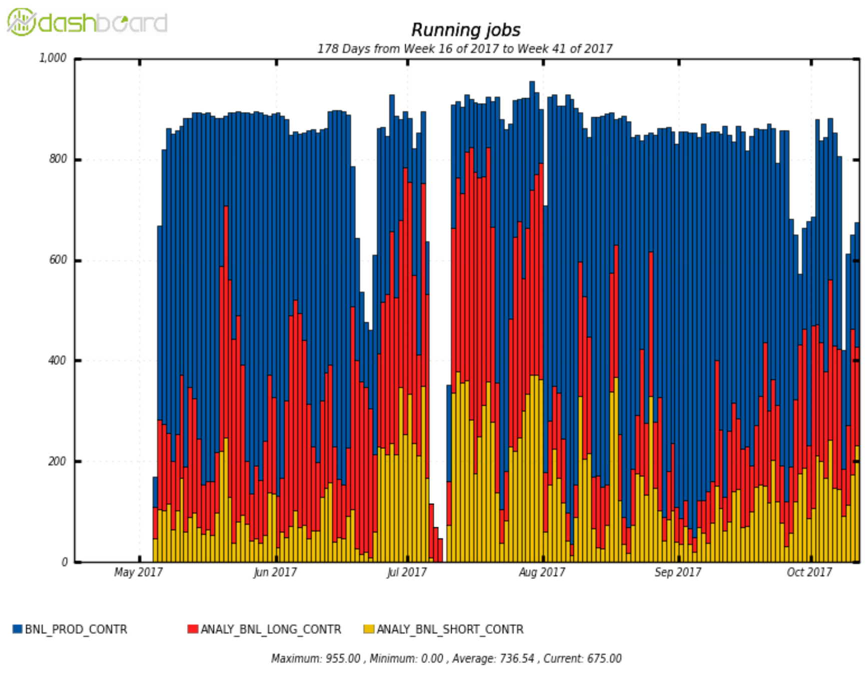


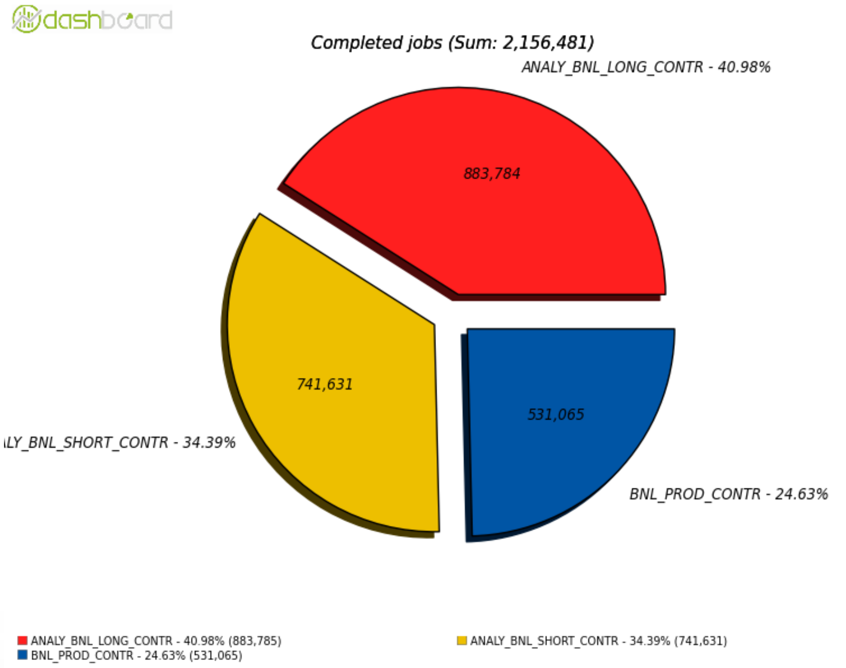
Time to run 'cat /etc/redhat-release' in a container

Singularity at RACF/SDCC - ATLAS

- Created a BNL ATLAS SL6 Singularity container image
  - Effectively contains minimum OS software dependency requirements for ATLAS
    - ~2 GB in size, but can be a sparse file with less than ~1.5 GB of actual used space
  - Custom image contains various site-specific bind directories
    - /pnfs, /home/condor, etc.
    - Singularity supports the use of OverlayFS (ENABLE_OVERLAY config option), which can be used to automatically create the needed directories
      - This was reportedly buggy with earlier SL7 kernels, but appears stable in recent tests with SL7.3
- Installed SL7 and Singularity on 30 (1 rack) ATLAS farm hosts, and put them in special HTCondor and Panda queues
  - Created a small 2-line HTCondor job wrapper (USER_JOB_WRAPPER config option) to force jobs which land on these hosts to execute in SL6 Singularity containers
  - Not using HTCondor's built-in Singularity support because we're using an older version which doesn't include it

- # Singularity at RACF/SDCC - ATLAS (Cont.)
  - After a successful period of running ATLAS HammerCloud tests in containers, we opened up the Singularity batch queues to real ATLAS production and analysis jobs
    - Container image available locally and in the SDCC CVMFS repo
    - Over 2 million jobs successfully completed since the queues were moved to production in late April





- Singularity at RACF/SDCC - ATLAS (Cont.)

  - ATLAS recently validated SL7 workernodes, but few sites have upgraded
  - https://twiki.cern.ch/twiki/bin/view/AtlasComputing/CentOS7Readiness
    - ATLAS doesn’t transparently support mixed SL6/SL7 farms at sites
    - Singularity offers a rolling/seamless OS upgrade path for our ATLAS workernodes - run ATLAS jobs in SL6 Singularity containers
      - SL6 support for new hardware will be an issue soon, and SL7 offers a number of useful features/improvements
      - New BNL ATLAS workernode purchase (set to arrive in November) of 90 nodes will be built with SL7: jobs will be executed in SL6 Singularity containers, using a configuration similar to our existing testbed
        - Systems will run jobs from the standard HTCondor/Panda queues used by our production workernodes
      - If no issues are encountered, the rest of our workernodes will be upgraded to SL7 in a rolling manner

- Singularity at RACF/SDCC - ATLAS (Cont.)
  - Many of our NFS mounts under /direct contain the '+' character – I.e /direct/usatlas+u
    - Singularity didn't permit this special character in bindmount pathnames
      - Submitted a patch to address this; merged in 2.3
  - Singularity is available for use on our IC and KNL HPC clusters, as well as our HEP/NP HTC resources
  - ATLAS is interested in making opportunistic use of these resources
    - CVMFS is not available on our HPC systems
    - Solution: creation of a “fat” ATLAS Singularity container image
      - Contains the entire contents of /cvmfs/atlas.cern.ch
      - ~600 GB in size
      - Single image file, allowing it to be easily shipped
      - Created by Wei Yang from SLAC
      - Stored in GPFS
    - Successfully ran test ATLAS jobs on KNL using this image

Singularity at RACF/SDCC – Other Experiments

- BNL RACF/SDCC is in the process of becoming the T1 computing center for BelleII in the US
  - Recently brought the first compute resources for BelleII at BNL online: 56 workernodes with SL7 installed
  - However, BelleII requires SL6 for their processing jobs
  - Adopted the same Singularity-based model from our ATLAS testbed
    - Created a BelleII workernode SL6 Singularity container image
      - Based on dockerfiles provided by PNNL
      - HTCondor jobs are executed in the BelleII singularity container, using the same HTCondor/Singularity configuration as our ATLAS testbed
- RHIC – Created an SL6 container image for our RHIC experiments
  - Compared to the minimal ATLAS and BelleII images, was a bit more complex to build, and larger: ~6.5 GB
    - Numerous non-stock/in-house developed packages
    - More heavily modified system configuration, and environment
  - In the early stages of testing; planned for use with CI

ATLAS SL6 Singularity Definition File

```
# BNL ATLAS SL6 Minimal Container
# Chris Hollowell <hollowec@bnl.gov>
BootStrap: yum
OSVersion: 6
MirrorURL: http://repo.bnl.gov/scientific/6x/x86_64/os/
Include: yum

%setup
    echo "Looking in directory '$SINGULARITY_ROOTFS' for /bin/sh"
    if [ ! -x "$SINGULARITY_ROOTFS/bin/sh" ]; then
        echo "Hrmm, this container does not have /bin/sh installed..."
        exit 1
    fi
    exit 0

%post
        yum -y groupinstall base
        cd /tmp
        wget http://linuxsoft.cern.ch/wlcg/sl6/x86_64/HEP_OSlibs_SL6-1.0.20-0.el6.x86_64.rpm
        yum -y install ./HEP_OSlibs_SL6-1.0.20-0.el6.x86_64.rpm
        mkdir -p /home/condor
        mkdir /cvmfs
        mkdir /afs
        mkdir -p /usatlas/u
        mkdir /etc/grid-security
        mkdir -p /pnfs/usatlas.bnl.gov
        ln -sfn /cvmfs/oasis.opensciencegrid.org/mis/certificates /etc/grid-security/certificates
        exit 0
```

- Singularity in Action

```
# Create a container
acas1801# singularity create /images/atlas_sl6.img
Initializing Singularity image subsystem
Opening image file: /images/atlas_sl6.img
Creating 2048MiB image
Binding image to loop
Creating file system within image
Image is done: /images/atlas_sl6.img
acas1801# singularity bootstrap /images/atlas_sl6.img /images/atlas_sl6.def
...

# Start a shell in the container
acas1801:~$ whoami
testuser
acas1801:~$ cat /etc/redhat-release
Scientific Linux release 7.3 (Nitrogen)
acas1801:~$ singularity shell /images/atlas_sl6.img
Singularity: Invoking an interactive shell within container...

Singularity.atlas_sl6.img> cat /etc/redhat-release
Scientific Linux release 6.9 (Carbon)
Singularity.atlas_sl6.img> whoami
testuser
```

- Future Plans

  - Running version 2.2.1 in production
    - Upgrade planned
      - Testing 2.3.2 (packages provided by OSG)
        - Some useful bug fixes, and new features
      - 2.4 was also recently released upstream
  - Currently storing container images locally, or as image files in our CVMFS repo (/cvmfs/sdcc.bnl.gov)
    - Plan to move all our images (except the ATLAS “fat” image) to directory-based container structures in CVMFS
  - Testing ATLAS' official images in CVMFS
  - ATLAS is working on developing Singularity support in Panda
    - Once this is completed, we can consider changing our configuration to allow the pilot to choose the containers to run jobs in