Quelling the Clamor for Containers

Vanessa Borcherding
Director, Scientific Computing Unit
Weill Cornell Medicine
Our Environment: High-Performance Computing for Biomedical Sciences

- Basic Science Research
  - “Integrated Systems Biology”
  - Molecular Dynamics
  - Genomics
  - Microscopy & Imaging

- A new emphasis on direct translational research
  - Clinicians are also Faculty and do basic research
  - One of our biggest projects is Precision Medicine

➢ Funding is primarily from NIH Grants and Philanthropy

Weill Cornell Medicine
Our resources, in a nutshell:

- 4000 dedicated compute cores, split across three major clusters and two smaller ones
  - Separate schedulers, a mix of SGE and SLURM
  - Mix of OS versions: CentOS 6/7

- ~5PB of storage, across many tiers
  - Including two GPFS clusters…

- Full Scientific Computing environment
What are Containers?
Virtual Machine’s lighter-weight little brother...
Why Containers?
Reproducibility.
Reproducibility.
Portability.
Reproducibility?
Reproducibility
Reproducibility
.... and that’s just the science.
Portability
.... But in reality, everyone’s environment is different.
Compiling Sucks

- It can be tricky - dependencies in particular.
- It can take forever - compiling and re-compiling the app and those pesky dependencies.
- Scientists have better things to do with their time.... like, say, Science.
- We help where we can, but some things are just too much to ask.
Realistically though, the number one question we get is some variant on “can I get sudo so that I can install this #!@& software?”
Say “no” enough times...
There’s gotta be a better way...
<Insert the choir of angels here....>
Our first foray into containers: Docker.
Docker for Precision Medicine

- Together with the Englander Institute for Precision Medicine, we helped support and build an end-to-end infrastructure in Docker including:
  - LIMS
  - Web Applications
  - Pipelines for sample analysis
- All of this is running under Docker, currently being transitioned seamlessly to HIPAA-compliant computing (which is possible *because* it’s Docker)

* Note: this is all self-contained - away from other users and shared storage
The inevitable question: “So, when are we getting Docker on the main cluster?”
You can run Docker on an HPC Cluster, but...

- Your scheduler needs to support it (i.e.: Univa) or you need to write a bunch of wrapper scripts for your scheduler
- Your nodes need to be running the latest versions of your OS across the board (ie: RHEL/CentOS 7)
- Security is an issue - Docker was built around a root-owned central process and escalation or running as root is built into the core concepts of the software; running as an unprivileged user is a relatively new feature
- To enforce running as an unprivileged user, you need to tweak SELINUX to only allow the root-owned processes you know to be safe
- Things like Infiniband support and GPFS support are doable (but not easy), but require privilege and careful work to get right
Containers built for HPC: Singularity

- Singularity shares as much of the host environment as possible - only encapsulates exactly what it needs to work. No full namespace encapsulation like Docker
- No route for escalation - if you want to be root in the container you must first be root outside of the container
- Much more forgiving about host kernels/libraries - I got this to run perfectly on CentOS 6 hosts
- All of your fancy filesystems and interconnects “just work” because it’s grabbing it from the host
- Either build your own images or import directly from Docker

http://singularity.lbl.gov
Building your Images

Your end users develop their image with sudo/root, entirely out of your cluster environment on their desktop platform of choice.
Watch the magic happen...

vanessa@node156 tmp $ cat /etc/centos-release
CentOS release 6.8 (Final)

vanessa@node156 tmp $ singularity shell ubuntu-xenial-tardis.img
Singularity: Invoking an interactive shell within container...

Singularity.ubuntu-xenial-tardis.img> cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=16.04
DISTRIB_CODENAME=xenial
DISTRIB_DESCRIPTION="Ubuntu 16.04 LTS"

Singularity.ubuntu-xenial-tardis.img> whoami
vanessa
Watch the magic happen...

```bash
Singularity.ubuntu-xenial-tardis.img> df -hl

Filesystem                        Size  Used  Avail  Use% Mounted on   
/dev/loop0                         3.9G   912M   2.8G   25%  /             
/dev/mapper/VG_node_dat01-tmplv    4.1T    7.8G   4.1T    1%  /tmp        
/dev/mapper/rootvg-rootlv         29G    2.2G   26G    8%  /etc/hosts   
/dev/sda1                          488M    70M   394M   15%  /boot        
/dev/athena                       1.7P   343T   1.4P   21%  /athena      
devtmpfs                          190G   220K   190G    1%  /dev        
tmpfs                             190G     0    190G   0%  /dev/shm     
/dev/mapper/rootvg-varlv          6.0G   661M   5.4G   11%  /var/tmp    
```
Watch the magic happen...

vanessa@node156  tmp $ singularity exec ubuntu-xenial-tardis.img tardis

In 900 years of time and space, I've never met anyone who wasn't important.

mrfast path: /usr/local/bin/mrfast
gnuplot path: /usr/bin/gnuplot

TARDIS: Toolkit for Automated and Rapid DIsovery of Structural variants.

Version 0.1-alpha

Last update: April 28, 2015, build date: Tue Mar 21 23:36:08 UTC 2017
AND THERE WAS MUCH REJOICING
“But it’s not Docker...”
Thank You!
Questions/Comments?
veb2005@med.cornell.edu

@neezbeez