Slurm Singularity Spank Plugin

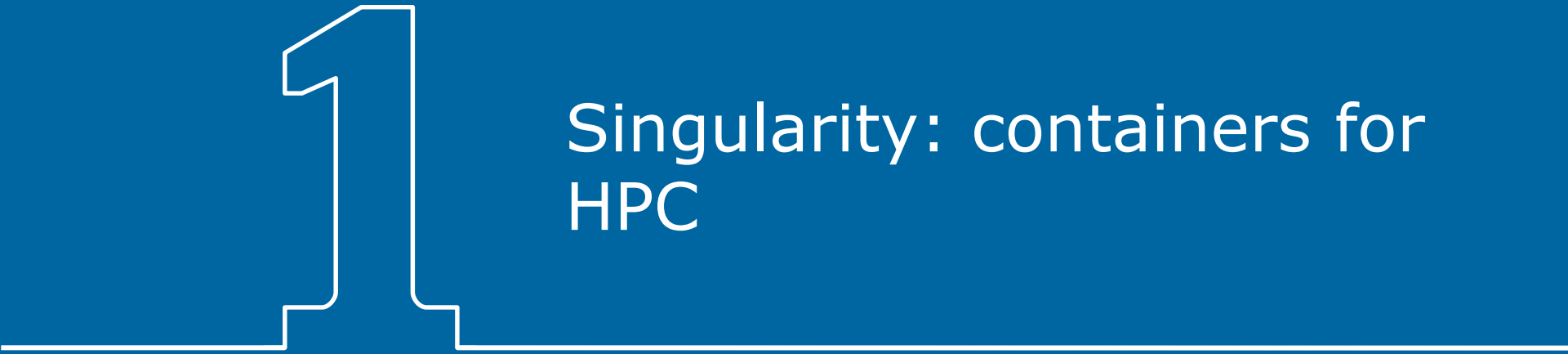
Martin Perry
Steve Mehlberg
Thomas Cadeau

26-09-2017

Bull
atos technologies

Slurm Singularity Spank Plugin

- Singularity overview
- Slurm SPANK Plugin for Singularity
- Singularity performance in HPC
- Conclusion



Singularity: containers for HPC

Singularity overview

- **What is Singularity?**
  - Open source, image-based Linux container technology providing encapsulated software environment to support software development and distribution.
  - Environment may include OS, file system, libraries, user applications and data.
  - Lightweight alternative to Virtual Machine.
  - Use cases: BYOE, reproducible science, software appliances, legacy code on old OSs, complicated software stacks or workflows.
  - More info: singularity.lbl.gov

Key Singularity commands

- Syntax: **singularity <command> <args>**
  - **create** Create a Singularity container.
  - **bootstrap** Bootstrap a Singularity build specification to build an image.
  - **exec** Run an executable (script, application, command)
  - **import** Import layers or other file content to your image.
  - **pull** Pull an image from Docker or Singularity Hub.
  - **run** Run image as an executable.
  - **shell** Launch interactive shell inside image.

- *Singularity through Slurm*
  - **srun** <srun options> singularity exec <container> <application> <options>
    - *$ srun -N2 singularity exec container.img cat /etc/redhat-release*

Singularity security features

- Security
  - Independent ownership of a container image and files within the image.
    - For example, image may be owned by a user but '/' within image must be owned by root.
  - No user contextual changes or root escalation allowed.
    - Calling user is maintained within the container. If you want to be root inside the container, you must first be root outside the container.
  - No root owned daemon processes
    - Preserves user privilege instead of inheriting root privilege of daemon.
- Access to HPC requirements
  - network, file system, standart IO, X11, MPI, pipes
- Portability
  - binary portable to any containers

SPANK for Singularity

Plugin overview

- Requirement
  - Provide a simple interface to allow Slurm users to run executables inside a Singularity container using srun.
- Status
  - Slurm provides host machine resource/workload management.
  - Singularity provides the software environment.
- Integration goal
  - Manage Singularity container images.
  - Generate required Singularity command from new srun options.

New srun options provided by the plugin

srun [--singularity-image=<container> [--singularity-env=<env script>] [--sgdebug]] [<other srun options>] <executable>

where:

<container> is the name of a Singularity container image in the SWRepo repository.

<env script> is an optional user script to set up the environment required to run <executable> inside <container> (environment variables, libraries, etc.)

--singularity-image may be abbreviated as --sgimage

--singularity-env may be abbreviated as --sgenv

Required software

- **Singularity**
  - Must be installed and configured on all nodes to be used by the plugin (login and compute nodes).
- **SWRepo**
  - Proprietary tool to manage Bull supercomputer system files (VMs, kernel files, initrd images, etc.).
  - Adapted for use by plugin as Singularity container image manager.
  - SWRepo server node used as central repository for Singularity container images.
  - Responsible for syncing container images to compute nodes as needed.
  - Configured with swrepo.conf and swrepo-server.conf.
- **rsync**
  - Fast, efficient Linux file copy/sync utility used by SWRepo.
- **The spank plugin itself**
  - Configured in the spank plugstack using sg_spank.conf.
- **Bash helper scripts**
  - Called by the plugin for basic file manipulation and command submission.

Plugin Configuration

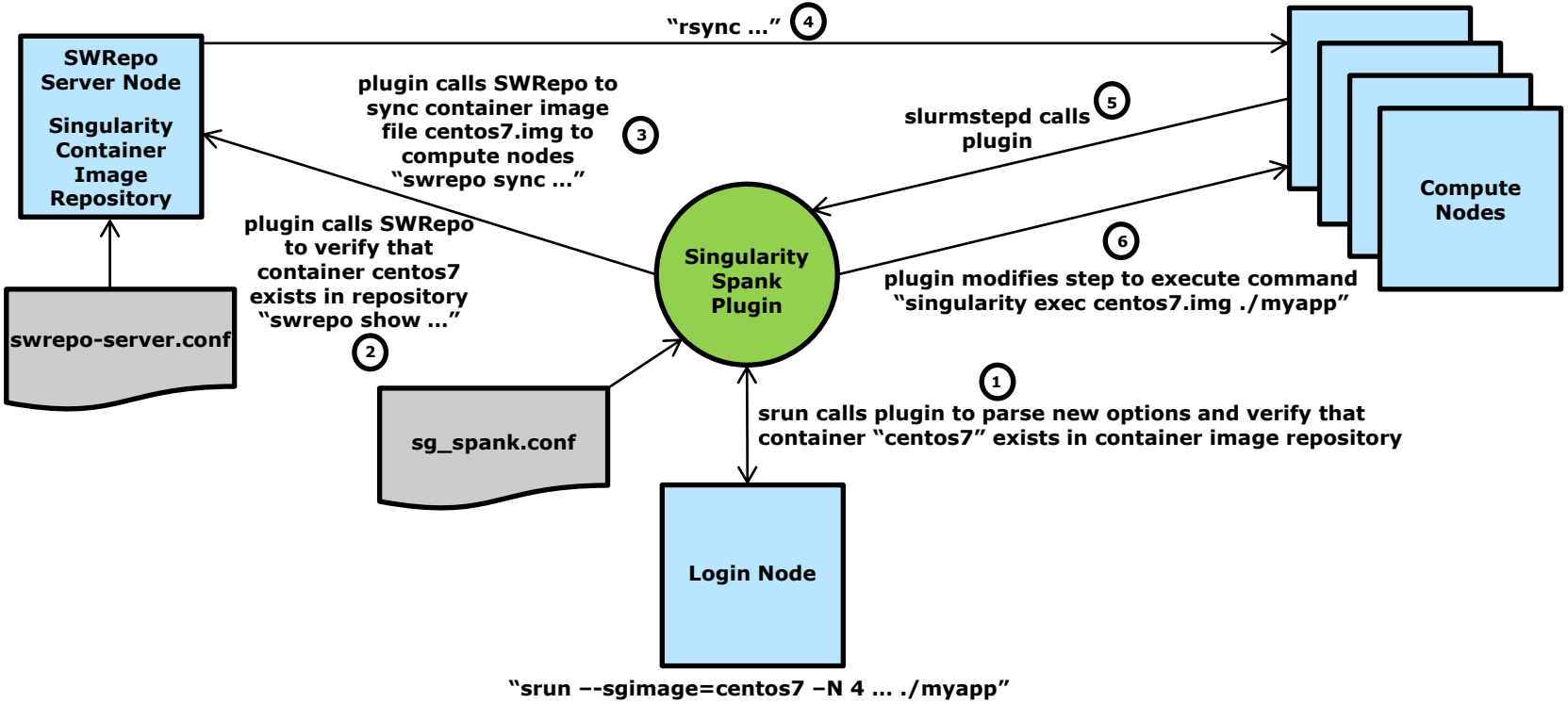
- **Example plugin configuration**

```
[trek0] (slurm) slurm> cat /etc/slurm/plugstack.conf
include /etc/slurm/plugstack.conf.d/*

[trek0] (slurm) slurm> ls /etc/slurm/plugstack.conf.d
sg_spank.conf

[trek0] (slurm) slurm> cat /etc/slurm/plugstack.conf.d/sg_spank.conf
################################################################################
# Singularity Slurm Spank Plugin Configuration file
#
# The following configuration parameters are available:
#
# swrepo_sync_dir : SWRepo sync directory (required; no default; must match swrepo-server.conf)
# working_dir : Working storage directory (optional; default is /var/tmp)
#
#-------------------------------------------------------------------------------
optional /usr/lib64/slurm/sg_spank.so swrepo_sync_dir=/var/lib/singularity
```

Architecture & Workflow



Example

[login0] $ srun -w compute1 cat /etc/system-release
Red Hat Enterprise Linux Server release 7.2 (Maipo)

[login0] $ srun -w compute1 singularity exec /path/centos7.img cat /etc/system-release
CentOS Linux release 7.2.1511 (Core)

[login0] $ srun -w compute1 --sgimage=centos7 cat /etc/system-release
CentOS Linux release 7.2.1511 (Core)

Performance tests

Testing Environment

- Four node cluster with 56 CPUs
  - Running Redhat 7.2, Open MPI 2.0.0, PMIX 1.1.4, Slurm 16.05.04
- Singularity 2.1 container
  - Running Centos 7, Open MPI 2.0.0, PMIX 1.1.4
- Benchmarks
  - Intel MPI Benchmarks (IMB)
  - High Performance LINPACK (HPL)
  - High Performance Conjugate Gradients (HPCG)
  - Nucleus for European Modeling of the Ocean (NEMO)

Slurm used to run benchmarks

- Bare-metal
  - $ srun –N4 -n56 -w compute[9-12] --mpi=pmix IMB-EXT.sh
  - $ srun –N4 –n16 -w compute[9-12] --mpi=pmix xhpcg.sh

- singularity
  - $ srun –N4 -n56 -w compute[9-12] --mpi=pmix --sgimage=centos7.img IMB-EXT.sh
  - $ srun –N4 –n16 -w compute[9-12] --mpi=pmix --sgimage=centos7.img xhpcg.sh

*.sh => executable (including path) with option

Results

- Each benchmark was run 10 times and the average was calculated for comparison.

Application	Nb nodes	Nb tasks	File system	Bare-Metal	Container
IMB-EXT	4	56	NFS:EXT4	14:47	14:42
IMB-MPI	4	56	NFS:EXT4	6:48	6:50
HPL short	4	56	NFS:EXT4	3:06	3:06
HPL long	4	16	NFS:EXT4	19:27	19:40
HPCG	4	16	NFS:EXT4	3:23	3:23
NEMO	2	16	NFS:Lustre	2:44	2:43

Conclusion

Summary & future work

- Perfomance summary:
  - Execution times, cpu and lustre usage are very comparable between singularity containers and bare-metal.
  - Using singularity containers in an HPC or Extreme Data Systems production environment should be as performant as using bare-metal itself.
- Full integration of versioning containers through Swrepo
  - Need to be careful on copy/sync containers
- Feedback from customer/community
  - need to be available

Thanks

For more information please contact:
thomas.cadeau@atos.net

Bull
atos technologies