Performance of LS-DYNA on Singularity Containers

Authors: Nirmala Sundararajan, Joshua Weage, Nishanth Dandapanthula

HPC Innovation Lab, February 2018

Overview

We often look at containers today and see the potential of accelerated application delivery, scaling and portability and wonder how we ever got by without them. This blog discusses the performance of LSTC LS-DYNA® within Singularity containers and on bare metal. This blog is the third in the series of blogs regarding container technology by the HPC Engineering team. The first blog Containers, Docker, Virtual Machines and HPC explored the use of containers. The second blog Containerizing HPC Applications with Singularity provided an introduction to Singularity and discussed the challenges of using Singularity in HPC environments. This third blog will focus on determining if there is a performance penalty while running the application (LS-DYNA) in a containerized environment.

Containerizing LS-DYNA using Singularity

LS-DYNA is a general-purpose finite element (FEA) program capable of simulating complex real world problems. It is used in multiple engineering disciplines such as automobile, aerospace, construction, military, manufacturing, and bioengineering industries. An application specific container is a lightweight bundle of an application and its dependencies. The first step in creating the container for LS-DYNA would be to have a minimal operating system (CentOS 7.3 in this case), basic tools to run the application and support for InfiniBand within the container. With this, the container gets a runtime environment, system tools, and libraries. The next step would be to install the application binaries within the container. The definition file used to create the LS-DYNA container is given below. In this file, the bootstrap references the kind of base to be used and a number of options are available. “shub” pulls images hosted on Singularity Hub, “docker” pulls images from Docker Hub and here yum is used to install Centos-7.

Definition File:

```
BootStrap: yum
OSVersion: 7
MirrorURL: http://vault.centos.org/7.3.1611/os/x86_64/
Include: yum


%post
# basic-tools and dev-tools
yum -y install evince wget vim zip unzip gzip tar perl
yum -y groupinstall "Development tools" --nogpgcheck

# InfiniBand drivers
yum -y --setopt=group_package_types=optional,default,mandatory groupinstall "InfiniBand
Support"
yum -y install rdma
yum -y install libibverbs-devel libsysfs-devel
yum -y install infinipath-psm

# Platform MPI
mkdir -p /home/inside/platform_mpi
```

```
chmod -R 777 /home/inside/platform_mpi
cd /home/inside/platform_mpi
/usr/bin/wget http://192.168.41.41/platform_mpi-09.1.0.1isv.x64.bin
chmod 777 platform_mpi-09.1.0.1isv.x64.bin
./platform_mpi-09.1.0.1isv.x64.bin -installdir=/home/inside/platform_mpi -silent
rm platform_mpi-09.1.0.1isv.x64.bin

# Application
mkdir -p /home/inside/lsdyna/code
chmod -R 777 /home/inside/lsdyna/code
cd /home/inside/lsdyna/code/usr/bin/wget http://192.168.41.41/ls-
dyna_mpp_s_r9_1_113698_x64_redhat54_ifort131_avx2_platformmpi.tar.gz
tar -xvf ls-dyna_mpp_s_r9_1_113698_x64_redhat54_ifort131_avx2_platformmpi.tar.gz
rm ls-dyna_mpp_s_r9_1_113698_x64_redhat54_ifort131_avx2_platformmpi.tar.gz

%environment
PATH=/home/inside/platform_mpi/bin:$PATH
LSTC_LICENSE=network
LSTC_MEMORY=auto
export PATH LSTC_L:wqICENSE_SERVER LSTC_LICENSE LSTC_MEMORY
%runscript
exec ls
/
```

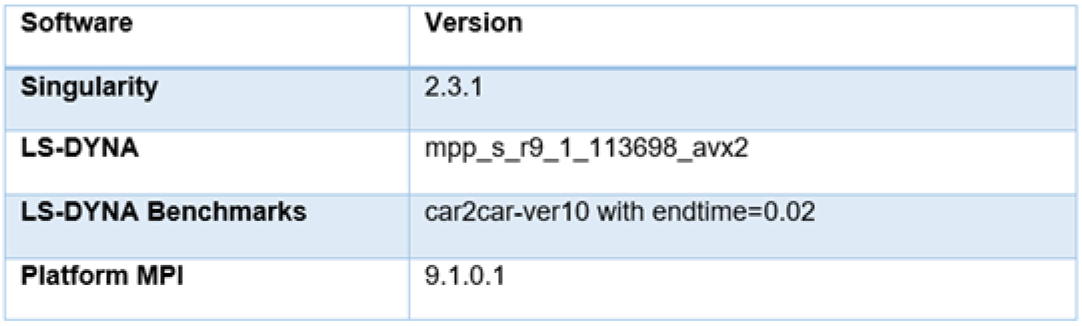
It is quick and easy to build a ready to use singularity image using the above definition file. In addition to the environment variables specified in the definition file, the value of any other variable such as LSTC_LICENSE_SERVER, can be set inside the container, with the use of the prefix “SINGULARITYENV_”. Singularity adopts a hybrid model for MPI support and the ‘mpirun’ is called, from outside the container, as follows:

```
/home/inside/platform_mpi/bin/mpirun -env_inherit -np 80 -hostfile hostfile singularity exec
completelsdyna.simg /home/inside/lsdyna/code/ls-
dyna_mpp_s_r9_1_113698_x64_redhat54_ifort131_avx2_platformmpi i=Caravan-V03c-2400k-main-
shell16-120ms.k memory=600m memory2=60m endtime=0.02
```

LS_DYNA car2car Benchmark Test

The car2car benchmark presented here is a simulation of a two vehicle collision. The performance results for LS-DYNA are presented by using the Elapsed Time metric. This metric is the total elapsed time for the simulation in seconds as reported by LS-DYNA. A lower value represents better performance. The performance tests were conducted on two clusters in the HPC Innovation Lab, one with Intel® Xeon® Scalable Processor Family processors (Skylake) and another with Intel® Xeon® E5-2600 v4 processors (Broadwell). The software installed on both the clusters was the same and is noted in Table 1

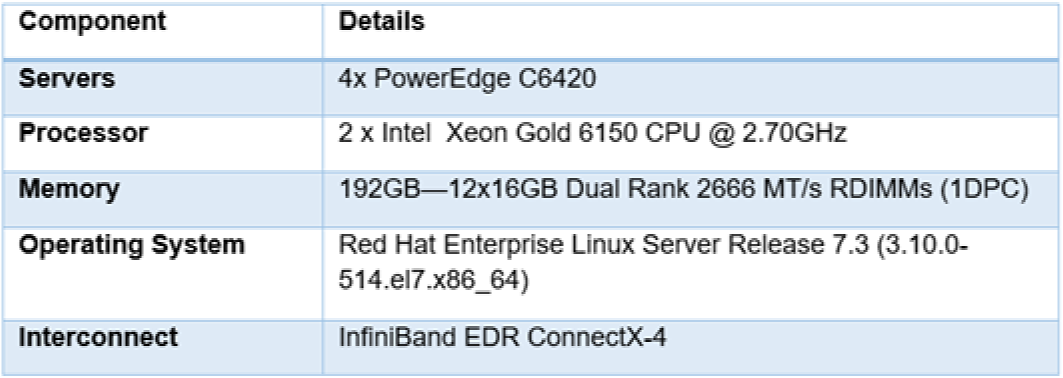
Table 1. Application Versions



Software	Version
Singularity	2.3.1
LS-DYNA	mpp_s_r9_1_113698_avx2
LS-DYNA Benchmarks	car2car-ver10 with endtime=0.02
Platform MPI	9.1.0.1

The configuration details of the Skylake cluster are shown below in Table 2.

Table 2. Skylake Cluster Configuration



Component	Details
Servers	4x PowerEdge C6420
Processor	2 x Intel Xeon Gold 6150 CPU @ 2.70GHz
Memory	192GB—12x16GB Dual Rank 2666 MT/s RDIMMs (1DPC)
Operating System	Red Hat Enterprise Linux Server Release 7.3 (3.10.0-514.el7.x86_64)
Interconnect	InfiniBand EDR ConnectX-4

Figure1 shows that there is no perceptible performance loss while running LS-DYNA inside a containerized environment, both on a single node and across the four node cluster. The results for the bare-metal and container tests were within 2% of each other, which is within the expected run-to-run variability of the application itself.

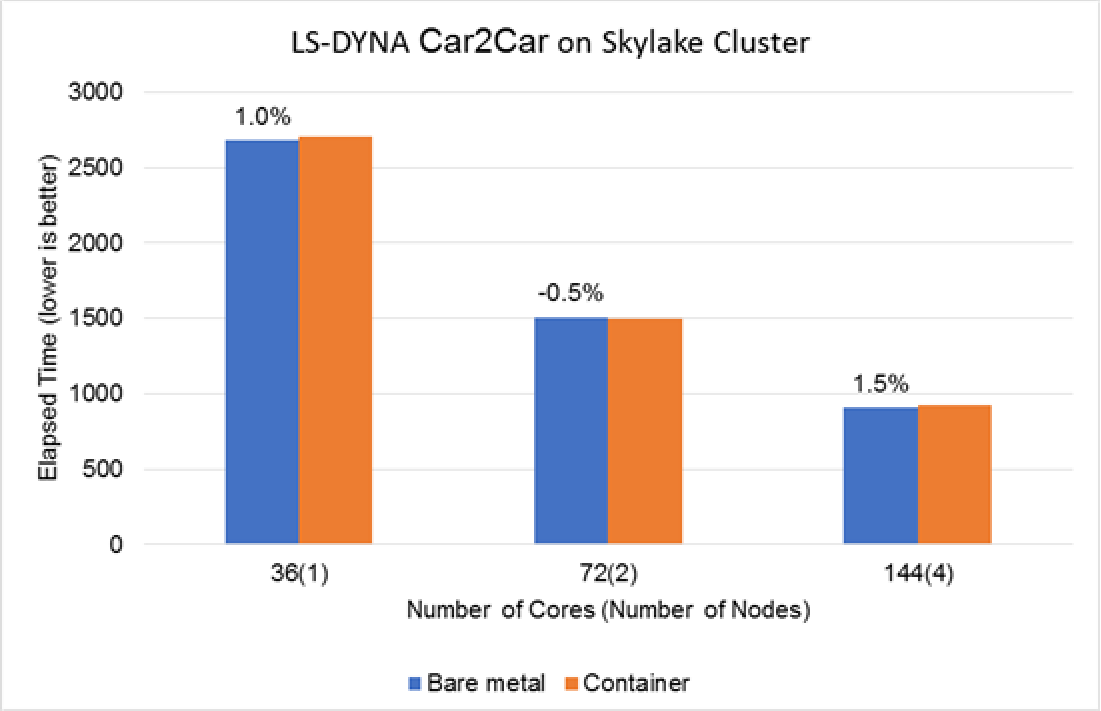
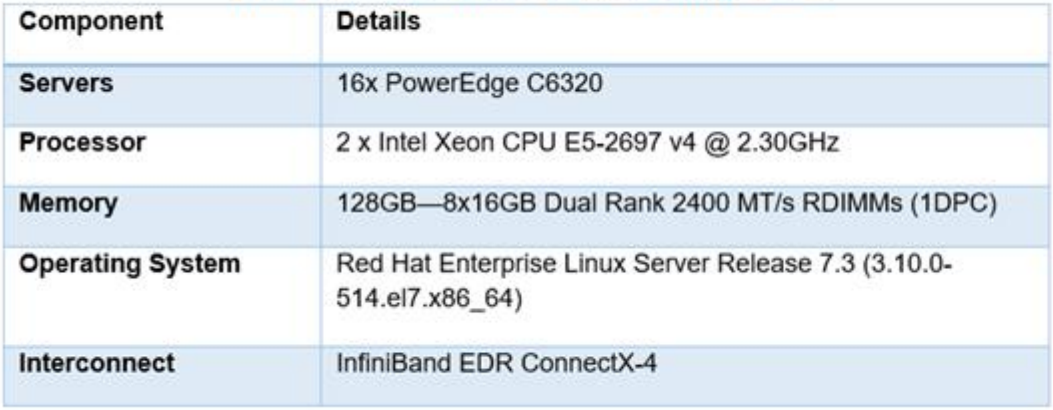


Figure 1. Performance inside singularity containers relative to bare metal on Skylake Cluster.

The configuration details of the Broadwell cluster are shown below in Table 3.

Table 3. Broadwell Cluster Configuration



Component	Details
Servers	16x PowerEdge C6320
Processor	2 x Intel Xeon CPU E5-2697 v4 @ 2.30GHz
Memory	128GB—8x16GB Dual Rank 2400 MT/s RDIMMs (1DPC)
Operating System	Red Hat Enterprise Linux Server Release 7.3 (3.10.0-514.el7.x86_64)
Interconnect	InfiniBand EDR ConnectX-4

Figure 2 shows that there is no significant performance penalty while running LS- DYNA car2car inside Singularity containers on Broadwell except for a six node run, where the bare metal configuration is better by up to 5.5%. This performance delta might be tolerable for those use cases where the benefits of using containers outweigh absolute performance considerations.

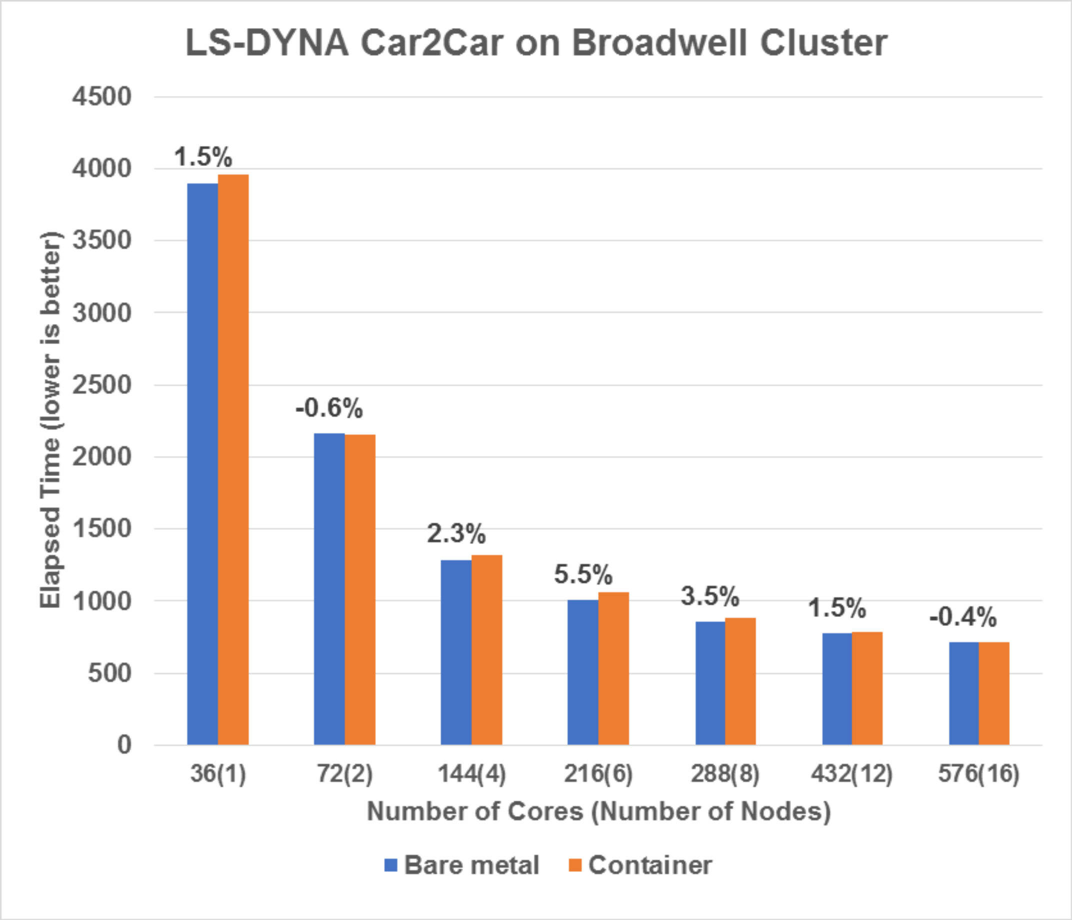


Figure 2. Performance inside singularity containers relative to bare metal on Broadwell Cluster.

Conclusion and Future Work

In this blog, we discussed how the performance of LS-DYNA within singularity containers is by and large at par with running the application on bare metal servers. The performance difference while running LS-DYNA within Singularity containers remains within 2%, for the most part, with the maximum variation at 5.5%. The Dell EMC HPC team will focus on containerizing other applications, and this blog series will be continued. So stay tuned for the next blog in this series!