ogs-container-maker: create and run OGS within portable Linux container

OGS Community Meeting 2019

Lars Bilke

Wednesday 27th March, 2019

Helmholtz Centre for Environmental Research GmbH – UFZ

Table of contents

1. Motivation and approach

2. ogs-container-maker

3. Usage

4. Outlook & conclussions

Motivation and approach

Motivation

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Challenges in configuration and setup of complex scientific simulation software

- Complex library and tool dependencies
- Different execution environments (workstation vs. HPC system)
- Dependencies for older software versions may not be available anymore

$\Longrightarrow$ Full reproducibility of scientific result data of simulations is a challenging task and may even be impossible.

Approach

Encapsulate code, dependencies, execution environment and data into a portable unit using container technologies

- Allow encapsulation of complex software setups
- Recent developments in scalable (HPC-enabled) container technologies, e.g. Singularity [1]
- Light-weight form of computer virtualization $\Longrightarrow$ High performance [2]

Objectives

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Outcome

- Provide portable containerized simulation software
  - Hiding complex software setups from the user
  - Executable on workstations (Linux, macOS) and arbitrary HPC systems
  - Integrate into typical scientific workflows
- Ensure reproducible simulation results
  - Signed and check-summed container image enhanced with meta-data about the software configuration
  - Archivable container images containing everything to reproduce results
  - Investigate long-term archival to meet regulations in safety-critical research areas

ogs-container-maker

Container technology: Singularity i

Singularity[1]

- Linux container technology, compatible to Docker images
- HPC-enabled by using regular schedulers (single executable)
- Easy to use, flexible workflows, image mobility
- Cryptographically signed immutable images (optional)

[1] http://singularity.lbl.gov

Container generation: definition

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Listing 1 : `example.def` – Example singularity definition file (recipe)[2]

```
1  Bootstrap: docker
2  From: ubuntu
3
4  %help
5    Help me. I'm in the container.
6
7  %post
8    apt-get update
9    apt-get install -y tool_x
10
11 %runscript
12   exec tool_x "$@"
```

[2] Singularity defintion file syntax:
`https://www.sylabs.io/guides/2.6/user-guide/container_recipes.html`

Container generation: definition (real-world example)

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

```
FROM ubuntu:17.10

# Generated with https://github.com/ufz/ogs-container-maker/commit/c0dc25d

RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        wget \
        tar \
        curl && \
    rm -rf /var/lib/apt/lists/*
# GNU compiler
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends software-properties-common && \
    apt-add-repository ppa:ubuntu-toolchain-r/test -y && \
    apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        gcc-6 \
        g++-6 \
        gfortran-6 && \
    rm -rf /var/lib/apt/lists/*
RUN update-alternatives --install /usr/bin/gcc gcc $(which gcc-6) 30 && \
    update-alternatives --install /usr/bin/g++ g++ $(which g++-6) 30 && \
    update-alternatives --install /usr/bin/gfortran gfortran $(which gfortran-6) 30 && \
    update-alternatives --install /usr/bin/gcov gcov $(which gcov-6) 30

# OGS base building block
# Python
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        python3 \
        libpython3-dev && \
    rm -rf /var/lib/apt/lists/*
# pip
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        python3-pip \
        python3-setuptools \
        python3-wheel && \
    rm -rf /var/lib/apt/lists/*
RUN pip3 install --upgrade pip
# pip
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        python3-pip \
        python3-setuptools \
        python3-wheel && \
    rm -rf /var/lib/apt/lists/*
RUN pip3 install virtualenv
# CMake version 3.13.4
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        wget && \
    rm -rf /var/lib/apt/lists/*
RUN mkdir -p /var/tmp && wget -q -nc --no-check-certificate -P /var/tmp https://cmake.org/files/v3.13/cmake-3.13.4-Linux-x86_64.sh && \
    /bin/sh /var/tmp/cmake-3.13.4-Linux-x86_64.sh --prefix=/usr/local --skip-license && \
    rm -rf /var/tmp/cmake-3.13.4-Linux-x86_64.sh
RUN apt-get update && \
    curl -s https://packagecloud.io/install/repositories/github/git-lfs/script.deb.sh | bash
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends software-properties-common && \
    apt-add-repository ppa:git-core/ppa -y && \
        apt-get update -y && \
        DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
            git \
            git-lfs \
            make \
            ninja-build && \
        rm -rf /var/lib/apt/lists/*
RUN git lfs install && \
    mkdir -p /apps /scratch /lustre /work /projects

# Package manager Conan building block
# pip
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        python3-pip \
        python3-setuptools \
        python3-wheel && \
    rm -rf /var/lib/apt/lists/*
RUN pip3 install conan==1.12.2
RUN mkdir -p /opt/conan && \
    chmod 777 /opt/conan
ENV CONAN_USER_HOME=/opt/conan
LABEL org.opengeosys.pm=conan \
      org.opengeosys.pm.conan.user_home=/opt/conan \
      org.opengeosys.pm.conan.version=1.12.2

# cppcheck version 1.87
RUN mkdir -p /var/tmp && wget -q -nc --no-check-certificate -P /var/tmp https://github.com/danmar/cppcheck/archive/1.87.tar.gz && \
    mkdir -p /var/tmp && tar -x -f /var/tmp/1.87.tar.gz -C /var/tmp -z && \
    mkdir -p /var/tmp/build && cd /var/tmp/build && cmake -DCMAKE_INSTALL_PREFIX=/usr/local/cppcheck -DCMAKE_BUILD_TYPE=RELEASE /var/tmp/cppcheck
                -1.87 && \
    cmake --build /var/tmp/build --target install -- -j$(nproc) && \
    rm -rf /var/tmp/1.87.tar.gz /var/tmp/build /var/tmp/cppcheck-1.87
ENV PATH=/usr/local/cppcheck/bin:$PATH

RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        doxygen \
        graphviz \
        texlive-base && \
    rm -rf /var/lib/apt/lists/*

# pip
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        python3-pip \
        python3-setuptools \
        python3-wheel && \
    rm -rf /var/lib/apt/lists/*
RUN pip3 install gcovr

# Package manager Conan building block
RUN apt-get update -y && \
    DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
        ccache && \
    rm -rf /var/lib/apt/lists/*
RUN mkdir -p /opt/cache && chmod 777 /opt/cache
ENV CCACHE_DIR=/opt/cache \
    CCACHE_MAXSIZE=15G \
    CCACHE_SLOPPINESS=pch_defines,time_macros
LABEL ccache.dir=/opt/cache \
      ccache.size=15G
```

Container generation: building image

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Turn the definition into an image:

```
1  sudo singularity build example.simg example.def
```

This creates the container image `example.simg`

hpc-container-maker [3] by NVidia[3]

- Generates container definitions from a high level Python recipe
- Contains a library of HPC building blocks
- Recipes become programable, e.g. generate multiple variations
- Creates Singularity and Docker definitions

[3] https://github.com/NVIDIA/hpc-container-maker

Listing 2 : `basic.py` – Install GNU compiler on Ubuntu

```
1 Stage0.baseimage('ubuntu:16.04')
2 Stage0 += packages(ospackages=['gcc', 'g++', 'gfortran'])
```

Listing 3 : Generate Singularity definition and build image

```
1 hpccm --recipe basic.py --format singularity > basic.def
2 sudo singularity build basic.simg basic.def
```

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

ogs-container-maker[4] builds upon hpc-container-maker and provides

- Building blocks for OGS requirements (e.g. VTK, PETSc)
- A parameterized combinatorial recipe for building OGS and all its requirements
- Scripts for building common OGS configurations

[4] https://github.com/ufz/ogs-container-maker

Listing 4 : Combinatorial options

```
All combinations of the given options will be generated

--format [{docker,singularity} [{docker,singularity} ...]]
--pm [{system,conan} [{system,conan} ...]]
                      Package manager to install third-party dependencies
                      (default: ['conan'])
--ompi [OMPI [OMPI ...]]
                      OpenMPI version, e.g. 2.1.1, 2.1.5, 3.0.1, 3.1.2
                      (default: ['off'])
--ogs [OGS [OGS ...]]
                      OGS GitHub repo in the form 'user/repo@branch' or
                      'off' to disable OGS building (default:
                      ['ufz/ogs@master'])
--cmake_args [CMAKE_ARGS [CMAKE_ARGS ...]]
                      CMake argument sets have to be quoted and **must**
                      start with a space. e.g. --cmake_args ' -DFIRST=TRUE
                      -DFOO=BAR' ' -DSECOND=TRUE' (default: [''])
```

ogs-container-maker: options ii

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Listing 5 : Build options

```
--build, -B           Build the images from the definition files (default:
                      False)
--upload, -U          Upload Docker image to registry (default: False)
--registry REGISTRY   The docker registry the image is tagged and uploaded
                      to. (default: registry.opengeosys.org/ogs/ogs)
--convert, -C         Convert Docker image to Singularity image (default:
                      False)
--runtime-only, -R    Generate multi-stage Dockerfiles for small runtime
                      images (default: False)
```

ogs-container-maker: options iii

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Listing 6 : Additional options

```
--base_image BASE_IMAGE
                      The base image. 'centos:7' is supported too. (default:
                      ubuntu:17.10)
--clang               Use clang instead of gcc (default: False)
--gui                 Builds the GUI (Data Explorer) (default: False)
--docs                Setup documentation requirements (Doxygen) (default:
                      False)
```

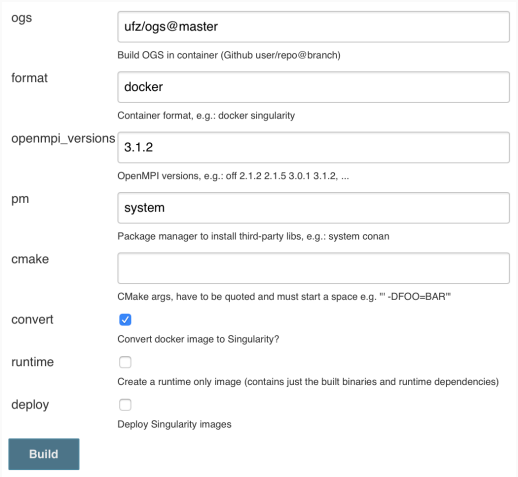
Usage

Container maker service on Jenkins

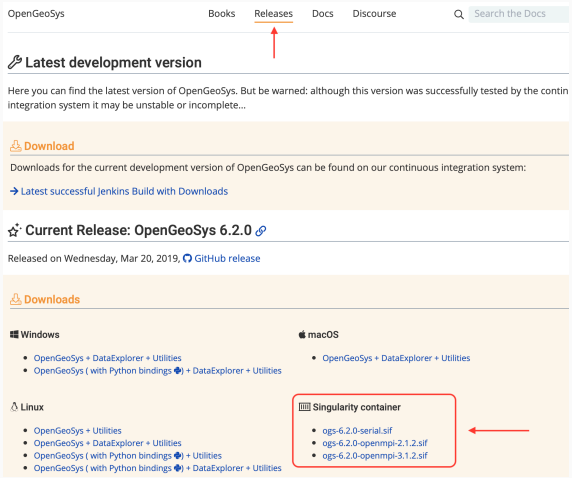
Available at `jenkins.opengeosys.org/job/ufz/job/ogs-container-maker/job/master/build`

- User defined container image generation
  - Specified OGS configurations are bundled into images
  - Image is generated by Jenkins
  - Generated images are temporarily archived on Jenkins for easy access

Container maker service on Jenkins



Download prebuilt container



Container usage

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

```
singularity exec ogs-6.2.0-serial.sif ogs some/path/project.prj
```

- Starts a container,
- mounts home directory inside the container,
- passes current working directory,
- runs OGS exe with passed project file.

Container usage

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

Listing 7 : Container usage examples

```
# Run serial benchmark with output validation (via vtkdiff)
singularity exec ogs-6.2.0-serial.sif ogs -o _out -r [ogs-sources]/Tests/Data/Mechanics/Linear [ogs-sources]/Tests/Data/Mechanics/Linear/disc_with_hole.prj
# Run parallel benchmark with MPI
mpirun -np 4 singularity exec ogs-6.2.0-openmpi-2.1.2.sif ogs -o _out_mpi [ogs-sources]/Tests/Data/Mechanics/Linear/disc_with_hole.prj
# Run other contained executables, e.g. vtkdiff
singularity exec ogs-6.2.0-serial.sif vtkdiff --help

# Shell into container
singularity shell ogs-6.2.0-serial.sif
# List files in the container
Singularity ogs-6.2.0-serial.sif:...> ls /usr/local/ogs/bin
... ogs tetgen vtkdiff
# Exit the container and get back to your hosts shell
Singularity ogs-6.2.0-serial.sif:...> exit
```

Outlook & conclussions

Outlook

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

- Check MPI compatibility (versions, host vs. container) on HPC environments
- Utilize multi-application / -configuration container with SCIF [4]
- Conduct HPC simulations on multiple environments
- Implement signing of containers
- Implement user feature requests ...

Conclusions

HELMHOLTZ CENTRE FOR ENVIRONMENTAL RESEARCH – UFZ

- Encapsulate OGS with all development and/or runtime dependencies in a single file (+ data)
- Use ogs-container-maker to generate arbitrary container definitions
- Use the Jenkins interface to generate container images
- Build the container once, run everywhere[5]

- Download prebuilt container: `opengeosys.org/releases`
- ogs-container-maker: `github.com/ufz/ogs-container-maker`

[5] ...has to be proven...

Thank you!

lars.bilke@ufz.de

References i

[1] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer.
Singularity: Scientific containers for mobility of compute.
PLOS ONE, 12(5):e0177459, May 2017.

[2] Carlos Arango, Rémy Dernat, and John Sanabria.
Performance Evaluation of Container-based Virtualization for High Performance Computing Environments.
arXiv preprint arXiv:1709.10140, 2017.

[3] Scott McMillan.
Making containers easier with hpc container maker.
In *In HPCSYSPROS18: HPC System Professionals Workshop*, Dallas, TX, November 2018.

[4] Vanessa Sochat.
The Scientific Filesystem.
GigaScience, 7(5), 03 2018.

References ii

ogs-container-maker: create and run OGS within portable Linux container, Lars Bilke