## Contents

1 Software Platform Procurement Procedures .......................... 3
  1.1 Chapter Summary and Highlights ........................................ 4
  1.2 Real-ESSI Program Executables Download and Install .................. 4
  1.3 Real-ESSI Simulator Install as Container through Docker ............... 5
    1.3.1 Real-ESSI Docker Image Development .............................. 5
    1.3.2 Running Real-ESSI Container through Docker ...................... 8
    1.3.3 Performance of Real-ESSI Container ............................... 10
  1.4 Real-ESSI Simulator System Install .................................... 10
    1.4.1 Student Manual for Real-ESSI Simulator System Install ............ 11
Chapter 1

Software Platform Procurement Procedures

(In collaboration with Dr. Han Yang and Mr. Hexiang Wang)
1.1 Chapter Summary and Highlights

1.2 Real-ESSI Program Executables Download and Install

Executables for the Real-ESSI Simulator program (Jeremić et al., 1988-2020) are available online. Pre-built executables are available for Linux, Ubuntu 18.04, and can be downloaded and installed by analyst.

In order for prebuild executables to be able to run on a user/analyst computer, system libraries have to be brought up to date and additional libraries installed. System libraries update/upgrade:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get dist-upgrade
sudo apt-get autoremove
```

For sequential and/or parallel version of Real-ESSI, additional libraries are needed, as described below:

**Sequential Version of Real-ESSI Program.** Libraries required to be installed for using sequential version of the Real ESSI program:

```
sudo apt-get install libboost-all-dev
sudo apt-get install libhdf5-dev
sudo apt-get install libtbb-dev
sudo apt-get install libssl1.0.0
```

**Parallel Version of Real-ESSI Program.** Libraries required to be installed for executing parallel version of the Real ESSI program:

```
sudo apt-get install libboost-all-dev
sudo apt-get install libhdf5-dev
sudo apt-get install libtbb-dev
sudo apt-get install mpich
sudo apt-get install libopenmpi-dev
sudo apt-get install libssl1.0.0
```

1.3 Real-ESSI Simulator Install as Container through Docker

Recent developments in virtualization of operating systems (OS) has created an opportunity to deploy programs and software systems as container images. Container images are used by the host OS (Linux, Windows, MacOS) to create a container. A container is a running instance of a container image, and is represented by a Linux/Windows/MacOS process that can be used to run programs that are installed within container. Programs that are installed within a container have all the necessary libraries available within container and are fully self sufficient, irrespective of what container host OS is used, be it Linux or Windows or MacOS.

More information used virtualization, containers, docker, etc. can be found at:

- [https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction/](https://developers.redhat.com/blog/2018/02/22/container-terminology-practical-introduction/)

Starting from Real-ESSI version 20.07, Real-ESSI Simulator is now available as a Docker Container Image, and can be installed and used on Linux, Windows and MacOS.

1.3.1 Real-ESSI Docker Image Development

This section is intended for Real-ESSI developers, users can skip this section. The development of Real-ESSI Docker image follows typical steps to 'dockerize' any application. Here are some very helpful sources:

- Official documentation: [https://docs.docker.com/](https://docs.docker.com/)
- A Docker Tutorial for Beginners: [https://docker-curriculum.com/#our-first-image](https://docker-curriculum.com/#our-first-image)
- How to dockerize any application: [https://hackernoon.com/how-to-dockerize-any-application-b60ad00e76da](https://hackernoon.com/how-to-dockerize-any-application-b60ad00e76da)
- Slimming Down Your Docker Images: [https://towardsdatascience.com/slimming-down-your-docker-images-275f0ca9337e](https://towardsdatascience.com/slimming-down-your-docker-images-275f0ca9337e)

It should be mentioned that there are many different ways and styles that can be employed to create Docker image. Here, multistage build is used to save build/debug time and, more importantly, reduce size of the final image.

Provided below are steps used to create the Real-ESSI Docker image.

- Obtain the source code of Real-ESSI.
- The following 'Dockerfile' is created to build the Real-ESSI Docker image.
FROM ubuntu:18.04 AS basesystem

MAINTAINER Han Yang <hhhyang@ucdavis.edu>

WORKDIR /usr/src

COPY . .

RUN useradd -m ubuntu && \
    apt-get update && apt-get install -y \
    bison \
    build-essential \
    cmake \
    flex \
    libboost-all-dev \
    libhdf5-serial-dev \
    liblapack-dev \
    libopenblas-dev \
    libopenmpi-dev \
    libpthread-workqueue-dev \
    libssl-dev \
    libtbb-dev \
    mpich \
    ssh \
    valgrind \
    wget \
    zlib1g-dev

FROM basesystem AS dependencies

RUN cd Real-ESSI && \
    mkdir -p ../RealESSI_Dependencies && \
    mkdir -p ../RealESSI_Dependencies/include && \
    mkdir -p ../RealESSI_Dependencies/lib && \
    mkdir -p ../RealESSI_Dependencies/bin && \
    mkdir -p ../RealESSI_Dependencies/SRC && \
    cd ../RealESSI_Dependencies && \
    ./build_libraries suitesparse && \
    ./build_libraries arpack && \
    ./build_libraries hdf5_sequential && \
    ./build_libraries tbb && \
    ./build_libraries lapack && \
    ./build_libraries parmetis && \
    ./build_libraries petsc_itself
FROM dependencies AS builder

RUN cd Real-ESI && \
    mkdir build && \
    cd build && \
    cmake .. && \
    make -j 16 && \
    cp essi essi_sequential && \
    cd .. && \
    mkdir pbuild && \
    cd pbuild && \
    cmake -DCMAKE_CXX_COMPILER=/usr/bin/mpic++ ← \
        -DPROGRAMMING_MODE=PARALLEL .. && \
    make -j 16 && \
    cp essi essi_parallel

FROM ubuntu:18.04

MAINTAINER Han Yang <hhhyang@ucdavis.edu>

RUN useradd -m ubuntu && \
    apt-get update && apt-get install -y \
    libboost-all-dev \
    libhdf5-dev \
    libopenmpi-dev \
    libtbb-dev \
    mpich \
    ssh

COPY --from=builder /usr/src/Real-ESI/build/essi_sequential ← \
    /usr/src/Real-ESI/pbuild/essi_parallel /usr/bin/

USER ubuntu

WORKDIR /workspace

VOLUME ["/workspace"]

- Put the 'Dockerfile' in the same directory with the source code of Real-ESI.
- Build the Real-ESI Docker image. This step usually takes a long time, especially for the first time.
  docker build -t realessilocal:test .
- Correctly tag your image. This is not only necessary for later push but also just a good practice to
organize your Docker images.

```
docker tag realessilocal:latest realessi/real-essi-repo:<tag>
```

Replace `<tag>` with the tag you want to use. It’s usually a version name.

- Push your build to Docker Hub. Make sure you have the proper permission to do so.

```
docker push realessi/real-essi-repo:<tag>
```

### 1.3.2 Running Real-ESSI Container through Docker

Provided below are steps needed to install and run Real-ESSI within a Docker Container. The following steps work for both Linux and Windows systems. In a Linux system, run the following commands in a terminal. In a Windows system, run these commands in PowerShell. It should also work for Mac OS but hasn’t been tested yet.

- Install Docker on the local computer, desktop, laptop. Documentation on how to install Docker on user OS can be found here:
  - Linux: [https://docs.docker.com/engine/install/#server](https://docs.docker.com/engine/install/#server)
  - Windows: [https://docs.docker.com/docker-for-windows/install/](https://docs.docker.com/docker-for-windows/install/)
  - MacOS: [https://docs.docker.com/docker-for-mac/install/](https://docs.docker.com/docker-for-mac/install/)

- Manage Docker as a non-root user on Linux hosts

  If you are using a Linux host, by default you need to run Docker using `sudo`. If you don’t want to preface the `docker` command with `sudo`, create a group called `docker` and add users to it.

  To create the `docker` group and add your user:

  1. Create the `docker` group.

     ```
sudo groupadd docker
     ```

     Sometimes the `docker` group might already exist after the installation of Docker. This is okay, just move on to the next step.

  2. Add your user to the `docker` group.

     ```
sudo usermod -aG docker $USER
     ```

     Replace `$USER` with your user name.
3. Log out and log back in so that your group membership is re-evaluated. On Linux, you can also run the following command to activate the changes to groups:

```
newgrp docker
```

4. Verify that you can run `docker` commands without `sudo`.

```
docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints an informational message and exits.

More information on managing Docker as a non-root user can be found here: https://docs.docker.com/engine/install/linux-postinstall/

- Pull the Real-ESSI image

```
docker pull realessi/real-essi-repo:tag
```

Replace `tag` with the latest version of Real-ESSI. For example, the latest version is 20.07, then the pull command is `docker pull realessi/real-essi-repo:20.07`.

Current Real-ESSI Simulator version is kept up to date at the Real-ESSI web site HERE. In addition, you can find tags of Real-ESSI at https://hub.docker.com/repository/docker/realessi/real-essi-repo/tags.

- Run the Real-ESSI image:

```
docker run -it --rm -v your_working_directory:/workspace realessi/real-essi-repo:tag
```

Again, replace `tag` with the version of Real-ESSI you pulled. Once you start running the Real-ESSI Docker image, you are working inside the container. The container is Ubuntu 18.04 with Real-ESSI installed. Note that you should replace `your_working_directory` with the absolute path of your working directory.

- Run Real-ESSI:

```
essi_sequential -f main.fei
```

Note that the current directory on your local machine is shared with the container, so it can work with any files there. The files need to have the correct permissions to be run by a non-administrator user. You can move files after the container started and they will be recognized by the container.
After the simulation is finished, simply exit the container. You will see the output files and log file in your current directory. They will not be erased when you exit the container.

### 1.3.3 Performance of Real-ESSI Container

To test the performance of Real-ESSI container, a series of sequential and parallel simulations are conducted. The results and comparison are summarized in Figure 1.1.

![Comparison of Real-ESSI performance on local Linux machine and Linux/Windows containers.](image)

Figure 1.1: Comparison of Real-ESSI performance on local Linux machine and Linux/Windows containers.

### 1.4 Real-ESSI Simulator System Install

In addition to the Real-ESSI Program, Real-ESSI Simulator system consists of a pre-processing modules and post-processing modules. Installation of pre-processing modules is described in Chapter 207, on page 1171 in Real-ESSI Executable Procurement Procedures version: December 3, 2020, 16:59.

Both pre and post processing manuals are also available through the main Real-ESSI Simulator web site: [http://real-essi.info/](http://real-essi.info/).

### 1.4.1 Student Manual for Real-ESSI Simulator System Install

Students at ETH, Mr. Max Sieber and Mr. Antonio Felipe Salazar created a manual for installation of the Real-ESSI Simulator system on virtual machine computers. The manual is available [HERE](http://real-essi.info/).
Bibliography
