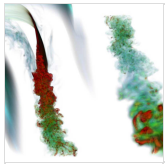

Jun 30, 2021

Run ReVisE from Docker

Alexey Kuzin[1], Stepan Orlov[1], Егор Усик[1], Alexey Zhuravlev[1]
[1]Peter the Great St.Petersburg Polytechnic University

 dx.doi.org/10.17504/protocols.io.bv5hn836

ReVisE

Alexey Kuzin
Peter the Great St.Petersburg Polytechnic University

ABSTRACT

This protocol describes run of ReVisE server from Docker container on Ubuntu. It describes retrieving of the Docker and the container only. It refers to another our protocol Prepare and run test on available dataset in order to provide possible workflow of ReVisE usage. This protocol is tested on Ubuntu 20.04 but it is not restricted by this OS version only.

DOI

dx.doi.org/10.17504/protocols.io.bv5hn836

PROTOCOL CITATION

Alexey Kuzin, Stepan Orlov, Егор Усик, Alexey Zhuravlev 2021. Run ReVisE from Docker . **protocols.io** https://dx.doi.org/10.17504/protocols.io.bv5hn836

KEYWORDS

ReVisE, visualization system

CREATED

Jun 26, 2021

LAST MODIFIED

Jun 30, 2021

PROTOCOL INTEGER ID

51081

MATERIALS TEXT

Visualization server

A computer (single node) equipped with one or more NVIDIA's GPU is required to run this protocol. The model of GPU may vary from GeForce 1060 to V100 or more recent.
The computer is expected to run the Linux operating system.
Minimum RAM requirement is 16 GB; one of the tests requires 128 GB RAM.
Free disk storage requirement is 210 GB to store all source and preprocessed datasets at once.
The computer is expected to have an Internet connection for downloading any of necessary prerequisites, and ReVisE Docker image.
If accessed remotely, the ssh connection is expected to work.
Super-user privileges might be necessary

Client machine

The client machine is expected to be a computer (desktop or laptop) having a display with FullHD resolution (1920 x 1080 pixels).
The client machine might be the same as the visualization server, or it can be a different computer connected to the visualization server remotely, via ssh.

BEFORE STARTING

The protocol assumes that the client machine is running Linux. However, this is not necessary if the client machine is used to access a remote visualization server. If a Windows machine is used, install PuTTY for establishing the ssh connection with the remote server.

The server machine must have
- NVIDIA GPU
- CUDA 11.1 or newer

The second requirement raises because we prepared Docker images for these CUDA versions only. If you have older CUDA version you still can use ReVisE, but you have to build it from sources, as it described in protocol Prepare and run test on available dataset

Docker installation

1 The machine where ReVisE server is to be run must have Docker Engine installed. One can install Docker Engine according to this reference. Or install Docker Community Edition (CE) according to the next sub-steps:

1.1 Remove old Docker packages, if they present:

```
sudo apt remove docker docker-engine docker.io containerd runc
```

1.2 Install some necessary packages:

```
sudo apt install apt-transport-https ca-certificates curl software-properties-
common
```

1.3 Add Docker repository to apt and install Docker:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
sudo apt update
```


Citation: Alexey Kuzin, Stepan Orlov, ÃÂÃÂ³ÃÂ¾ÃÂ ÃÂ£ÃÂÃÂ¸ÃÂº, Alexey Zhuravlev (06/30/2021). Run ReVisE from Docker . https://dx.doi.org/10.17504/protocols.io.bv5hn836

```
sudo apt install -y docker-ce
```

1.4 Test installation

```
sudo docker run hello-world
```

2 Install NVIDIA container runtime.

```
curl -s -L https://nvidia.github.io/nvidia-container-runtime/gpgkey | sudo apt-key add -
```

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
```

```
curl -s -L https://nvidia.github.io/nvidia-container-runtime/$distribution/nvidia-container-runtime.list | sudo tee /etc/apt/sources.list.d/nvidia-container-runtime.list
```

```
sudo apt update
```

```
sudo apt install nvidia-container-runtime
```

Restart docker after the installation:

```
sudo systemctl stop docker
sudo systemctl start docker
```

ReVisE container loading and startup

3 Open Linux terminal. This step is essential if ReVisE container is to be started on remote computer. In this case one needs to connect to remote computer via ssh with the next command:

```
ssh -L 1234:hostname:1234 -L 3000:hostname:3000 username@hostname
```

Therefore ports 1234 and 3000 should be forwarded.

Otherwise, if ReVisE server is used locally, use usual Linux terminal.

4 Load Docker image. This step should be done once on the first run of the container. Now we provide Docker images for CUDA 11. 1 and CUDA >= 11.2
CUDA 11.1:

```
sudo docker pull deadmorous/revise:cuda11.1
```

CUDA 11.2 and newer:

```
sudo docker pull deadmorous/revise:v1
```

5 Docker image startup:

```
sudo docker run --gpus all -ti --rm -v <HOME_PATH>:<HOME_PATH> -p 3000:3000 -p 1234:1234 <IMAGE>  /bin/bash
```


Citation: Alexey Kuzin, Stepan Orlov, ÃÂÃÂ³ÃÂ¾ÃÂ ÃÂ£ÃÂÃÂ¸ÃÂº, Alexey Zhuravlev (06/30/2021). Run ReVisE from Docker . https://dx.doi.org/10.17504/protocols.io.bv5hn836

Where <HOME_PATH> is home directory, for example, /home/username.
<IMAGE> - the Docker image name used: deadmorous/revise:v1 or deadmorous/revise:cuda11.1

In the case of successful startup you will be redirected to the image's console.

ReVisE server startup

6 By this time ReVisE server is ready to be run from Docker container. It is similar to startup of usual ReVisE installation. Therefore, the workflow of the protocol Prepare and run test on available dataset is applicable since the step 3. First of all set ReVisE environment:

```
source scripts/env.sh
```

7 Go to the directory where datasets are to be installed.

```
cd data
```

8 Now you can perform all measurements after step 4 from protocol Prepare and run test on available dataset. The Docker container has the same datasets that the usual ReVisE installation contains.

Citation: Alexey Kuzin, Stepan Orlov, ÃÂÃÂ³ÃÂ¾ÃÂ ÃÂ£ÃÂÃÂ¸ÃÂº, Alexey Zhuravlev (06/30/2021). Run ReVisE from Docker . https://dx.doi.org/10.17504/protocols.io.bv5hn836