



# Demonstration of Docker on macOS

LarSoft Coordination Meeting

Pengfei Ding

January 30<sup>th</sup>, 2017

# Introduction

- I am showing a quick demonstration of using docker on macOS;
- This may provide us another way to support different platforms.
- The latest docker version works with cvmfs on macOS out of the box.
- If you have cvmfs installed on the macOS, pull down this docker image: **dingpf/slf6.7**
- Files to build this docker image: <http://home.fnal.gov/~dingpf/slf6.7-docker.tar.gz>
- This image has authentication utilities and osg toolkits installed; Kerberos, voms client, xrdcp etc. work out of the box;
- This image has X11 and vnc (and noVNC) installed; so user can run EventDisplay inside it.
- It works with the latest docker version from mac OSX El Captain, and does not require System-Integrity-Protection to be disabled.
- This image can be used by all IF experiments.

# Demo -- running "lar" jobs

```
host node $ docker run --rm -it -v /cvmfs:/cvmfs:cached -v $HOME:/home/me \
dingpf/slf6.7
```

- In the docker container, run

```
Container $ source /cvmfs/dune.opensciencegrid.org/products/\
dune/setup_dune.sh
```

```
Container $ setup dunetpc v06_60_00_02 -q e14:prof
```

```
Container $ lar -c prodsingle dune35t.fcl
```

```
TrigReport ----- Event Summary -----
TrigReport Events total = 1 passed = 1 failed = 0
```

```
TrigReport ----- Modules in End-Path: end_path -----
TrigReport Trig Bit#          Run      Success      Error Name
TrigReport      0      0            1            1            0 out1
```

```
TimeReport ----- Time Summary ---[sec]-----
TimeReport CPU = 10.380000 Real = 12.275191
```

```
MemReport ----- Memory Summary ---[base-10 MB]-----
MemReport VmPeak = 1184.91 VmHWM = 589.718
```

```
Graphics systems deleted.
Visualization Manager deleting...
Art has completed and will exit with status 0.
bash-4.1$
bash-4.1$
bash-4.1$
bash-4.1$ hostname
61207fe85e56
```

# Demo -- running compiling dunetpc

- With the following alias:

```
host node $ alias build_dune="docker run --entrypoint \"/home/me/compile.sh\" --rm --user 0 -v /cvmfs:/cvmfs -v $HOME:/home/me dingpf/slf6.7"
```

- compile.sh looks like:

```
#!/bin/bash
source /cvmfs/dune.opensciencegrid.org/products/\
dune/setup_dune.sh
cd $my_dev
source $my_dev/localProducts_XXXX/setup
make install -j4
```

- You can edit the code on host machine; then run “build\_dune” to do the compiling.

# Demo -- running dune EVD

```
host node $ docker run --rm -it -p 5900:5900 -p 6900:6900 -v \
  /cvmfs/:/cvmfs:cached -v $HOME:/home/me dingpf/slf6.7
```

- In the docker container, run

```
Container $ start-xvnc.sh
```

- Connect to VNC via browser or VNC viewer.

```
http://localhost:6900
```

```
Or vnc://localhost:5900
```

- Run the following

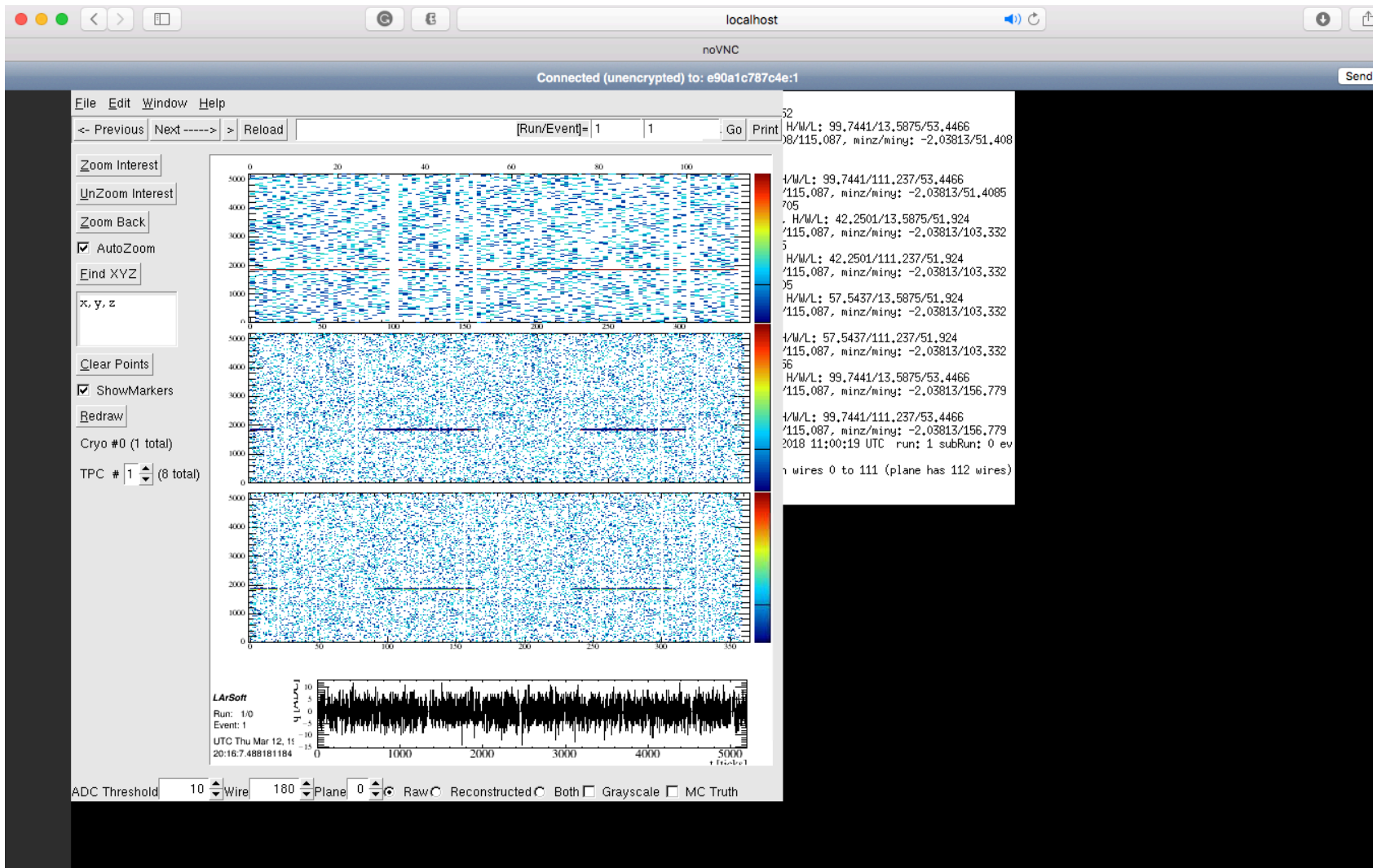
```
Container $ source /cvmfs/dune.opensciencegrid.org/products/\
dune/setup_dune.sh
```

```
Container $ setup dunetpc v06_60_00_02 -q e14:prof
```

```
Container $ lar -c prodsingle_dune35t.fcl
```

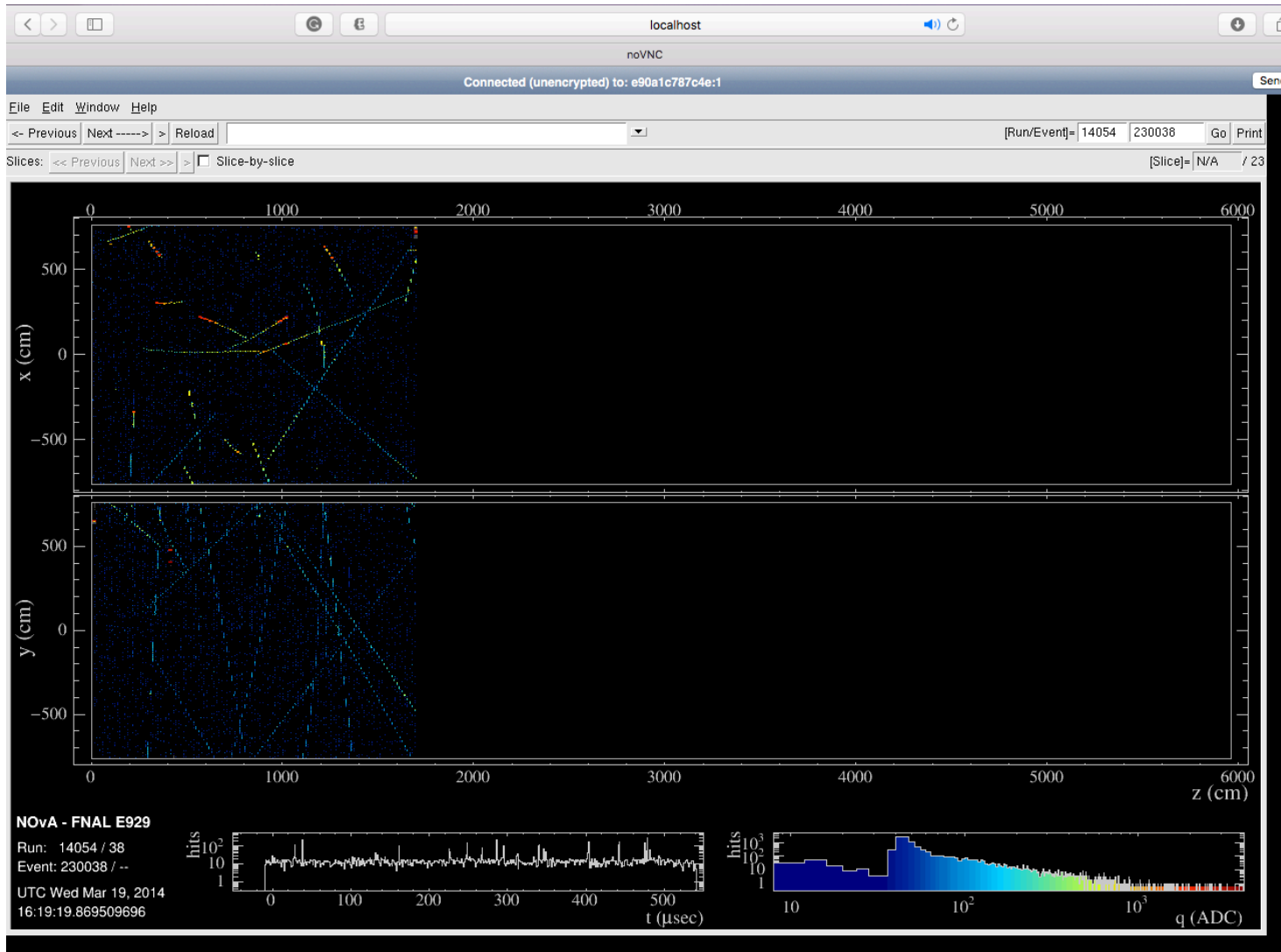
```
Container $ lar -c evd_dune35t.fcl single35t_gen.root
```

# Demo -- running dune EVD (cont'd)





# Demo -- running NOvA EVD



# Limitations

- cvmfs on macOS relies on osxfuse;
- **The latest docker version has performance issues with osxfuse;**
- This can be resolved by the following solution with the same docker image (tested with cvmfs installed in VM, shared by docker/singularity containers, did not test using cvmfs on macOS):
  - A. Vagrant + virtualbox + docker
  - B. Vagrant + virtualbox + singularity
- **Solution B is preferable. We need singularity to utilize GPU in HPC.**



# To-do list

- Test with mac IDEs, e.g. Xcode, CLion etc:
  - Editing
  - Building
  - Debugging
  - Profiling
  
- Adding GPU support, need to use singularity containers.
  - Access GPU on Linux should work
  - May require SIP to be disabled on macOS
  - And, as of version 1.2, TensorFlow no longer provides GPU support on macOS.

# Discussion

- Three possible docker image production lines:
  - **Generic docker image** shared by all experiments(like the one in the demo), **provides experiment collaborators quick access to a development environment**
  - Experiment specific image:
    - Contains one version of experiments software
    - New version needs to be made when a new release is cut
    - Can make a "**development**" and "**production**" version for each release;
      - **Development version contains full stack of software, can be used on NERSC or anywhere else where CVMFS is not available**
      - **Production version might be an optimized stripped down version designed for being used in scaled by production**
      - **Can set up Jenkins to do nightly builds for docker images?**

# Backup

- Install cvmfs on macOS
- Follow instructions on:  
<https://cernvm.cern.ch/portal/filesystem/quickstart>
- wget <http://home.fnal.gov/~dingpf/cvmfs.tar.gz>
- Copy the configuration files in the above tarball to /etc/cvmfs