Build JEDI environment with Docker

Docker installation

- Docker Community Edition for **Mac**, please refer to here.
- Docker Community Edition for **Windows**, please refer to here.
- Docker Community Edition for **Linux**:
  - Docker for **CentOS** distribution, please refer to here.
  - Docker for **Ubuntu** distribution, please refer to here.
  - Docker for **Debian** distribution, please refer to here.
  - Docker for **Fedora** distribution, please refer to here.
  - ...

> **Docker cheat sheets**
>
> - https://coderwall.com/p/2es5jw/docker-cheat-sheet-with-examples
> - https://github.com/wsargent/docker-cheat-sheet
> - https://www.docker.com/sites/default/files/Docker_CheatSheet_08.09.2016_0.pdf

Step-by-step guide

1. ### Download Docker image for JEDI

Download JEDI Docker image

```
[xinzhang@localhost jedi]$ docker container rm $(docker container ls -aq)     #  Clean up all the running
/dead/sleep container


[xinzhang@localhost jedi]$ docker image rm -f $(docker image ls -aq)  # clean up all cached images


[xinzhang@8-236 ~]$ docker pull jcsda/jedi-docker:gnu7  # check out the JEDI Docker image from the
registry
gnu7: Pulling from jcsda/jedi-docker
ae79f2514705: Pull complete
5ad56d5fc149: Pull complete
170e558760e8: Pull complete
395460e233f5: Pull complete
6f01dc62e444: Pull complete
fb1cc11fd1c7: Pull complete
58bde7425ccf: Pull complete
0ee8a8ab489f: Pull complete
66182f50c1b8: Pull complete
8c5e9c726feb: Pull complete
47feb6bfe9f4: Pull complete
03cc9d13057b: Pull complete
Digest: sha256:c0441594a97f339e5c9d79da7551c7a4ee81254d835d1035e2bc5f9fc656235a
Status: Downloaded newer image for jcsda/jedi-docker:gnu7


[xinzhang@8-236 ~]$ docker image ls    # list the available local Docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
jcsda/jedi-docker   gnu7               bf752acacf92        18 minutes ago      1.72GB
```

⭐ if you want to know more about the jedi-docker images, please refer to here.

---

2. ### Check out the GSI code and get case (T24) data (outside of container)

check out the GSI code

```
[xinzhang@8-236 ~] mkdir -p ~/jedi/code


[xinzhang@8-236 ~] cd ~/jedi/code

[xinzhang@8-236 code]$ git clone https://Your_UserName@bitbucket.org/jcsda/gsi-vlab.git gsi  # Check out
the GSI code, please replace Your_UserName with your bitbucket username

[xinzhang@8-236 code]$ cd gsi/

[xinzhang@8-236 gsi]$ git submodule init   # Check out the external repositories
Submodule 'fix' (git@bitbucket.org:jcsda/fix) registered for path 'fix'
Submodule 'libsrc' (git@bitbucket.org:jcsda/external-libs) registered for path 'libsrc'

[xinzhang@8-236 gsi]$ git submodule update   # Check out the external repositories
Cloning into 'fix'...
remote: Counting objects: 332, done.
remote: Compressing objects: 100% (235/235), done.
remote: Total 332 (delta 102), reused 320 (delta 96)
Receiving objects: 100% (332/332), 961.58 MiB | 357.00 KiB/s, done.
Resolving deltas: 100% (102/102), done.
Checking connectivity... done.
Submodule path 'fix': checked out 'd13e5a23dde538ceefedc39746b029878ee6ff60'
Cloning into 'libsrc'...
remote: Counting objects: 896, done.
remote: Compressing objects: 100% (630/630), done.
remote: Total 896 (delta 263), reused 890 (delta 260)
Receiving objects: 100% (896/896), 5.57 MiB | 1.59 MiB/s, done.
Resolving deltas: 100% (263/263), done.
Checking connectivity... done.
Submodule path 'libsrc': checked out '8fa69e1fd011b658de6cd2e9f08cf1ad77566f4f'

[xinzhang@8-236 gsi] cd fix


[xinzhang@8-236 gsi] git checkout rev1
```

macOS Users

The default macOS file system is HFS+ and, during installation, it is installed as **case-insensitive** by default. With Docker for Mac, file systems are shared from macOS into containers in **the same way as** they operate in macOS. As a result, if a file system on macOS is case-insensitive that behavior is shared by any bind mount from macOS into a container.

A workaound is needed to compile GSI code in container on macOS:

workaround for macOS

```
[xinzhang@localhost jedi]$ cd ~/jedi/code/gsi


[xinzhang@localhost gsi]$ mv libsrc/bufr/bufrlib.PRM libsrc/bufr/bufrlib.JEDI


[xinzhang@localhost gsi]$ vi core-libs/bufr/CMakeLists.txt   # make following modification,
replace '.PRM' with '.JEDI'


[xinzhang@localhost gsi]$ git diff core-libs/bufr/CMakeLists.txt
diff --git a/core-libs/bufr/CMakeLists.txt b/core-libs/bufr/CMakeLists.txt
index a597f69..ae45fcc 100644
--- a/core-libs/bufr/CMakeLists.txt
+++ b/core-libs/bufr/CMakeLists.txt
@@ -3,12 +3,12 @@ cmake_minimum_required(VERSION 2.6)
 if(BUILD_CORELIBS)
   file(GLOB BUFR_F77_SRC ${BUFR_DIR}/*.f ${BUFR_DIR}/*.F)
   file(GLOB BUFR_C_SRC ${BUFR_DIR}/*.c)
-  file(GLOB BUFR_PRM ${BUFR_DIR}/*.PRM)
+  file(GLOB BUFR_PRM ${BUFR_DIR}/*.JEDI)

   ADD_CUSTOM_COMMAND( OUTPUT "${CMAKE_INCLUDE_OUTPUT_DIRECTORY}/bufrlib.prm"
      PRE_BUILD
-     COMMAND cpp -P -D_REAL8_ -DWRF -DLINUX -DPGI -traditional-cpp ${BUFR_DIR}/bufrlib.PRM -o
${CMAKE_INCLUDE_OUTPUT_DIRECTORY}/bufrlib.prm
-     DEPENDS ${BUFR_DIR}/bufrlib.PRM
+     COMMAND cpp -P -D_REAL8_ -DWRF -DLINUX -DPGI -traditional-cpp ${BUFR_DIR}/bufrlib.JEDI -o
${CMAKE_INCLUDE_OUTPUT_DIRECTORY}/bufrlib.prm
+     DEPENDS ${BUFR_DIR}/bufrlib.JEDI
      )
   add_custom_target(bufrlib_prm DEPENDS ${CMAKE_INCLUDE_OUTPUT_DIRECTORY}/bufrlib.prm )
   if( BUFR_F77_SRC )
```

Get case (T24) data

```
[xinzhang@8-236 jedi]$ cd ~/jedi


[xinzhang@8-236 jedi]$ wget ftp://ftp.ucar.edu/pub/mmm/xinzhang/caseT24.tar.gz


If you cannot run wget, try curl and skip the next two steps:
curl ftp://ftp.ucar.edu/pub/mmm/xinzhang/caseT24.tar.gz | tar xvz


[xinzhang@8-236 jedi]$ tar xvf caseT24.tar.gz


[xinzhang@localhost jedi]$ rm caseT24.tar.gz


[xinzhang@localhost jedi]$ ls -la
total 1163160
drwxrwxr-x. 5 xinzhang xinzhang         64 Oct  4 12:11 .
drwxrwxr-x. 4 xinzhang xinzhang         33 Oct  4 11:30 ..
drwxrwxr-x. 3 xinzhang xinzhang         17 Oct  4 11:30 code
drwxrwxr-x. 4 xinzhang xinzhang         36 Oct  3 13:34 data
drwxrwxr-x. 2 xinzhang xinzhang         32 Oct  3 13:55 test
```

List of Case (T24) Content

```
1) T24 background(size: 52M):

On Theia: /scratch3/BMC/wrfruc/mhu/jedi/casedata/T24_2016030406/gest24/
On Docker: ~/jedi/data/case/T24_2016030406/gest24

aircraft_t_bias.gdas.2016030400  sfcf05.gdas.2016030400  sigf04.gdas.2016030400
biascr.gdas.2016030400           sfcf06.gdas.2016030400  sigf05.gdas.2016030400
biascr_pc.gdas.2016030400        sfcf07.gdas.2016030400  sigf06.gdas.2016030400
radstat.gdas.2016030400          sfcf08.gdas.2016030400  sigf07.gdas.2016030400
sfcf03.gdas.2016030400           sfcf09.gdas.2016030400  sigf08.gdas.2016030400
sfcf04.gdas.2016030400           sigf03.gdas.2016030400  sigf09.gdas.2016030400


2) T24 ensemble forecast(size:178M):

On Theia:/scratch3/BMC/wrfruc/mhu/jedi/casedata/T24_2016030406/enst24
On Docker: ~/jedi/data/case/T24_2016030406/enst24

20 member, hourly output during 3-h to 09-h forecast initialized from 2016030400
  sfg_2016030400_fhr03s_mem020
  sfg_2016030400_fhr04s_mem020
  sfg_2016030400_fhr05s_mem020
  sfg_2016030400_fhr06s_mem020
  sfg_2016030400_fhr07s_mem020
  sfg_2016030400_fhr08s_mem020
  sfg_2016030400_fhr09s_mem020

3) Observations:
On Theia:/scratch3/BMC/wrfruc/mhu/jedi/casedata/T24_2016030406/obs
On Docker: ~/jedi/data/case/T24_2016030406/obs

3.1) Original observation file: 1bamua.gdas.2016030406, prepbufr.gdas.2016030406


3.2) Small observation file: 1bamua.gdas_picked, prepbufr_picked

List of observations in prepbufr_listobs.txt:
  header: number, level number, SID XOB YOB DHR TYP ELV SAID T29
  observation:POB QOB TOB ZOB UOB VOB PWO CAT PRSS
  obs quality: PQM QQM TQM ZQM WQM NUL PWQ
  obs error: POE QOE TOE NUL WOE NUL PWE

List of observations in 1bamua_listobs.txt:
  header: number, channel number, SAID FOVN YEAR MNTH DAYS HOUR MINU SECO CLAT CLON CLATH CLONH HOLS
  header 2: SAZA SOZA BEARAZ SOLAZI
  obs (channel):
```

sample code to read NetCDF observation file

```
1) code location:

gsi/util/read_ncobs

2) compile the code:

make

3) executable is: read_ncobs.exe
It will read BUFR radiance data from sample file:amsua_n19_wprofiles.nc4
```

3. Build GSI code and run GSI case

suppose you already checked out the GSI code as $HOME/jedi/code/gsi and have test case directory as $HOME/jedi/data.

Build GSI code and run GSI case

```
[xinzhang@8-236 jedi]$ cd ~


[xinzhang@8-236 ~] mkdir -p ~/jedi/build/gsi   # Prepare the directory for GSI building, out-of-source
building

[xinzhang@8-236 ~]$ docker run -it --rm -v $(pwd)/jedi:/jedi jcsda/jedi-docker   # Start a container
instance from the image, mount local $(pwd)/jedi to /jedi of container.

root@c959e2137acc:/usr/local$ cd /jedi/build/gsi     # Enter into directory for GSI building

root@339677624bb2:/jedi/build/gsi$ export FC=mpif90

root@339677624bb2:/jedi/build/gsi$ cmake -DBUILD_CORELIBS=ON -DUSE_WRF=OFF -DBUILD_GLOBAL=ON /jedi/code
/gsi

root@339677624bb2:/jedi/build/gsi$ make -j`nproc`  # parallel compilation with the availble cores

root@339677624bb2:/jedi/build/gsi$ ls -al bin/
total 44560
drwxr-xr-x. 2 root root       44 Sep 29 17:24 .
drwxr-xr-x. 9 1000 1001      251 Sep 29 17:17 ..
-rwxr-xr-x. 1 root root  8266248 Sep 29 17:24 enkf_gfs.x
-rwxr-xr-x. 1 root root 37358984 Sep 29 17:23 gsi_global.x

root@339677624bb2:/jedi/code/gsi/build$ cd /jedi/test

root@339677624bb2:/jedi/test$ ./run_gsi_picked.csh
```

mpirun in container as root

in Docker container, user has **root** access, **mpirun/mpiexec may reports following error**:

mpirun in container as root

```
root@8d47b01cda3e:~$ mpirun
--------------------------------------------------------------------------
mpirun has detected an attempt to run as root.
Running at root is *strongly* discouraged as any mistake (e.g., in
defining TMPDIR) or bug can result in catastrophic damage to the OS
file system, leaving your system in an unusable state.

You can override this protection by adding the --allow-run-as-root
option to your cmd line. However, we reiterate our strong advice
against doing so - please do so at your own risk.
```

as the message shows, please add --allow-run-as-root after mpirun/mpiexe as:

```
root@8d47b01cda3e:~$ mpirun --allow-run-as-root -np 4 ./a.out
```

Read Diag (O-B) information from GSI test case

```
After a successful GSI run, we can check the O-B for each observation by reading diagnosis files.

1) combine subdomain drag files to one drag file:
 #cd /jedi/test/testT24_picked
 #ls dir.*
 #cat dir.0000/conv_01 dir.0001/conv_01 > conv_ges
```

```
 #cat dir.0000/amsua_metop-b_01 dir.0001/amsua_metop-b_01 > amsua_metop-b_ges

2) compile read_daig utilities:
 #/jedi/code/gsi/util/Analysis_Utilities/read_diag
 #make
Should generate two executables: read_diag_conv.exe read_diag_rad.exe


3) read conventional observations:
  #vi namelist.conv
It should be set up like this:
 &iosetup
  infilename='/jedi/test/testT24_picked/conv_ges',
  outfilename='./results_conv_ges',
 /

  # ./read_diag_conv.exe
Now, we should see a text file: results_conv_ges.
The content of each column is listed below
    1-3: variable name,station ID,itype,
    4-7: obs relative time,latitude,longitude,pressure,
    8-14:iuse,observation,background, O-B (for wind, U and then V)

4) read radiance drag:

  #vi namelist.conv
It should be set up like:
 &isotope
  infilename='/jedi/test/testT24_picked/conv_ges',
  outfilename='./results_conv_ges',
 /

  # ./read_diag_rad.exe
Now, we should see a new text file: results_amsua_metop-b_ges
The content of this text file is listed below:

head information (number 1-26):
    1:observation latitude (degrees)
    2:observation longitude (degrees)
    3:model (guess) elevation at observation location
    4:observation time (hours relative to analysis time)
    5:sensor scan position
    6:satellite zenith angle (degrees)
    7:satellite azimuth angle (degrees)
    8:solar zenith angle (degrees)
    9:solar azimuth angle (degrees)
   10:sun glint angle (degrees) (sgagl)
   11:fractional coverage by water
   12:fractional coverage by land
   13:fractional coverage by ice
   14:fractional coverage by snow
   15:surface temperature over water (K)
   16:surface temperature over land (K)
   17:surface temperature over ice (K)
   18:surface temperature over snow (K)
   19:soil temperature (K)
   20:soil moisture OR graupel water path (if gmi .or. saphir)
   21:surface land type
   22:vegetation fraction OR scattering index from AMSU-A (if radmod%lcloud_fwd .and. sea)
   23:snow depth OR integrated CLWP (kg/m**2) from background (if radmod%lcloud_fwd .and. sea)
   24:surface wind speed (m/s)
   25:cloud fraction (%) OR if microwave:
      cloud liquid water (kg/m**2) OR
         cloud liquid water (kg/m**2) if(radmod%lcloud_fwd .and. sea)
         clw (kg/m**2) from retrievals if(gmi .or. amsr2)
   26:cloud top pressure (hPa) OR if microwave:
      total column precip. water (km/m**2) OR
         retrieved CLWP (kg/m**2) from simulated BT if((radmod%lcloud_fwd .and. sea) .or. gmi .or. amsr2)

channel information (loop through each channel with column list), for each column:
   1: observed brightness temperature (K)
```

```
   2: observed - simulated Tb with bias correction (K)
   3: observed - simulated Tb with no bias correction (K)
   4: inverse observation error
   5: quality control mark or event indicator
   6: surface emissivity
   7: stability index
   8: indicator of cloudy consistency (if lcloud_fwd) OR d(Tb)/d(Ts)
```

---

4. ## Check out the OOPS code (outside of container)

check out the OOPS code

```
[xinzhang@localhost jedi]$ cd ~/jedi/code


[xinzhang@localhost code]$ git clone -b feature/ufo https://github.com/UCAR/oops.git   # Check out the
oops code to oops directory
```

---

5. ## Build and test OOPS code

Build the OOPS code

```
[xinzhang@localhost code]$ cd ~

[xinzhang@localhost ~] mkdir -p ~/jedi/build/oops   # Prepare the directory for OOPS building, out-of-
source building

[xinzhang@localhost ~]$ docker container run -it --rm -v $(pwd)/jedi:/jedi jcsda/jedi-docker  # Start
the container and mount local $(pwd)/OOPS to /OOPS of container

root@3cfbdc6da2b0:/usr/local$ cd /jedi/build/oops

root@3cfbdc6da2b0:/jedi/build/OOPS$ ecbuild --build=release -DLAPACK_LIBRARIES=$LAPACK_LIBRARIES /jedi
/code/oops

root@3cfbdc6da2b0:/jedi/build/OOPS$ make -j`nproc`
...
...
Scanning dependencies of target test_qg_4densvar.x
[ 99%] Built target test_qg_4densvar.x
[ 99%] Linking CXX executable test_qg_increment
[ 99%] Built target test_qg_increment
[100%] Linking CXX executable test_qg_localization
[100%] Built target test_qg_localization

root@3cfbdc6da2b0:/jedi/build/OOPS/oops$ ctest
...
...
108/108 Test #108: test_qg_dfi ......................   Passed    0.46 sec

100% tests passed, 0 tests failed out of 108

Label Time Summary:
boost         =   1.13 sec (53 tests)
executable    =   1.16 sec (56 tests)
fortran       =   0.04 sec (3 tests)
oops          =  23.89 sec (108 tests)
script        =  22.73 sec (52 tests)

Total Test time (real) =  23.94 sec
```

---

6. ## Download WRF and WPS (outside of container)

Download WPS/WRF

```
[xinzhang@localhost jedi]$ cd ~/jedi/code

[xinzhang@localhost code]$ wget http://www2.mmm.ucar.edu/wrf/src/WPSV3.9.1.TAR.gz

[xinzhang@localhost code]$ tar xvf WPSV3.9.1.TAR.gz

[xinzhang@localhost code]$ rm -f WPSV3.9.1.TAR.gz


[xinzhang@localhost code]$ wget http://www2.mmm.ucar.edu/wrf/src/WRFV3.9.1.1.TAR.gz

[xinzhang@localhost code]$ tar xvf WRFV3.9.1.1.TAR.gz

[xinzhang@localhost code]$ rm -f WRFV3.9.1.1.TAR.gz

If you cannot run wget, try curl:
     curl http://www2.mmm.ucar.edu/wrf/src/WRFV3.9.1.1.TAR.gz | tar xvz
     curl http://www2.mmm.ucar.edu/wrf/src/WPSV3.9.1.TAR.gz | tar xvz




[xinzhang@localhost code]$ ls -la
total 12
drwxrwxr-x.  6 xinzhang xinzhang   53 Oct  5 16:01 .
drwxrwxr-x. 13 xinzhang xinzhang  196 Oct  5 14:45 ..
drwxrwxr-x. 13 xinzhang xinzhang  243 Oct  5 14:29 gsi
drwxrwxr-x. 10 xinzhang xinzhang 4096 Oct  5 11:13 oops
drwxr-xr-x.  7 xinzhang xinzhang 4096 Aug 17 11:10 WPS
drwxr-xr-x. 17 xinzhang xinzhang 4096 Aug 28 15:01 WRFV3
```

---

7. ## Build WRF and WPS code

Build WRF and WPS code

```
[xinzhang@localhost ~]$ docker container run -it --rm -v $(pwd)/jedi:/jedi jcsda/jedi-docker  # Start
the container and mount local $(pwd)/OOPS to /OOPS of container

root@acb8561561ca:/usr/local$ cd /jedi/code/WRFV3

root@acb8561561ca:/jedi/code/WRFV3$ ./configure  # Select 34 for dmpar gnu

root@acb8561561ca:/jedi/code/WRFV3$ vi configure.wrf   # Becasue WRF has problem to test mpi2 support,
please add -DMPI2_SUPPORT after DM_CC = mpicc

root@acb8561561ca:/jedi/code/WRFV3$ ./compile em_real  # For unknown reason, module_ra_rrtmg_swf.f90
might fail to be compiled due to internal compiler error, typing "./compile em_real" again usually can
solve the problem.
...
...
==========================================================================
build started:   Thu Oct  5 22:12:18 UTC 2017
build completed: Thu Oct 5 22:21:58 UTC 2017

--->                  Executables successfully built                  <---

-rwxr-xr-x. 1 root root 36472488 Oct  5 22:21 main/ndown.exe
-rwxr-xr-x. 1 root root 36346432 Oct  5 22:21 main/real.exe
-rwxr-xr-x. 1 root root 35944328 Oct  5 22:21 main/tc.exe
-rwxr-xr-x. 1 root root 40051304 Oct  5 22:21 main/wrf.exe

==========================================================================


root@acb8561561ca:/jedi/code/WRFV3$ cd ../WPS


root@e7c4293f629c:/jedi/code/WPS$ ./configure   # Select 3 for dmpar gfortran


root@acb8561561ca:/jedi/code/WPS$ vi configure.wrf   # edit COMPRESSION_LIBS and COMPRESSION_INC as
following lines and remove -f90=gfortran from DM_FC and remove -cc=gcc from DM_CC
...
...
...
#
#   Settings for Linux x86_64, gfortran    (dmpar)
#
#
COMPRESSION_LIBS    = -L/usr/local/lib -ljasper -lpng -lz
COMPRESSION_INC     = -I/usr/local/include
FDEFS               = -DUSE_JPEG2000 -DUSE_PNG
SFC                 = gfortran
SCC                 = gcc
DM_FC               = mpif90
DM_CC               = mpicc
...
...
...


root@acb8561561ca:/jedi/code/WPS$ compile


root@acb8561561ca:/jedi/code/WPS$ ls -al *.exe
lrwxrwxrwx. 1 root root 23 Oct  6 16:02 geogrid.exe -> geogrid/src/geogrid.exe
lrwxrwxrwx. 1 root root 23 Oct  6 16:02 metgrid.exe -> metgrid/src/metgrid.exe
lrwxrwxrwx. 1 root root 21 Oct  6 16:02 ungrib.exe -> ungrib/src/ungrib.exe
```

8. ## Download MPAS code (outside of container)

Download MPAS code

```
[xinzhang@localhost code]$ git clone -b v5.2 https://github.com/MPAS-Dev/MPAS-Release.git
```

9. ## Build MPAS code

Build MPAS code

```
[xinzhang@localhost MPAS-Release]$ cd ~

[xinzhang@localhost ~]$ docker container run -it --rm -v $(pwd)/jedi:/jedi jcsda/jedi-docker  # Start
the container and mount local $(pwd)/OOPS to /OOPS of container

root@acb8561561ca:/usr/local$ cd /jedi/MPAS-Release

root@acb8561561ca:/jedi/code/MPAS-Release$ vi ./src/core_atmosphere/physics/checkout_data_files.sh #
Replace the MPAS-Data address to https

root@acb8561561ca:/jedi/code/MPAS-Release$ git diff
diff --git a/src/core_atmosphere/physics/checkout_data_files.sh b/src/core_atmosphere/physics
/checkout_data_files.sh
index e62b466..f2c3cba 100755
--- a/src/core_atmosphere/physics/checkout_data_files.sh
+++ b/src/core_atmosphere/physics/checkout_data_files.sh
@@ -56,7 +56,7 @@ fi
 which git
 if [ $? -eq 0 ]; then
    echo "*** trying git to obtain WRF physics tables ***"
-   git clone git://github.com/MPAS-Dev/MPAS-Data.git
+   git clone https://github.com/MPAS-Dev/MPAS-Data.git
    if [ $? -eq 0 ]; then
       mv MPAS-Data/atmosphere/physics_wrf/files/* physics_wrf/files
       rm -rf MPAS-Data

root@acb8561561ca:/jedi/code/MPAS-Release$ make gfortran CORE=atmosphere
...
...
make[2]: Leaving directory '/home/xinzhang/jedi/code/MPAS-Release/src/core_atmosphere'
*****************************************************************************
MPAS was built with default double-precision reals.
Debugging is off.
Parallel version is on.
Papi libraries are off.
TAU Hooks are off.
MPAS was built without OpenMP support.
MPAS was built with .F files.
The native timer interface is being used
Using the PIO 1.x library.
*****************************************************************************
make[1]: Leaving directory '/home/xinzhang/jedi/code/MPAS-Release'
```

10. ## Test MPAS

Test MPAS

```
root@acb8561561ca:/jedi/code/MPAS-Release$ wget ftp://ftp.ucar.edu/pub/mmm/xinzhang/mpas_10242_case.tgz


root@acb8561561ca:/jedi/code/MPAS-Release$ tar xvf mpas_10242_case.tgz



root@acb8561561ca:/jedi/code/MPAS-Release$ mpirun --allow-run-as-root -np 8 atmosphere_model # or mpirun
--allow-run-as-root -np 1 atmosphere_model

task 0 of 8 is running
task 1 of 8 is running
task 2 of 8 is running
task 3 of 8 is running
task 6 of 8 is running
task 7 of 8 is running
task 4 of 8 is running
task 5 of 8 is running
```

---

11. Check out the FV3 code (outside of container)

check out the FV3 code

```
[xinzhang@localhost jedi]$ cd ~/jedi/code

[xinzhang@localhost code]$ git clone https://Your_UserName@bitbucket.org/jcsda/comfv3.git comfv3   #
Check out the FV3 code, please replace Your_UserName with your bitbucket username

[xinzhang@localhost code] cd comfv3

[xinzhang@localhost comfv3] git submodule init

[xinzhang@localhost comfv3] git submodule update

[xinzhang@localhost comfv3] git branch -a

[xinzhang@localhost comfv3] git checkout feature/gfortran_build

[xinzhang@localhost comfv3] cd FV3

[xinzhang@localhost FV3] git checkout feature/gfortran_build

[xinzhang@localhost FV3] cd ../NEMS

[xinzhang@localhost NEMS] git checkout feature/gfortran_build
```

---

12. Build the FV3 code

Build FV3 code

```
[xinzhang@localhost NEMS]$ cd ~

[xinzhang@localhost ~]$ docker container run -it --rm -v $(pwd)/jedi:/jedi jcsda/jedi-docker  # Start
the container and mount local $(pwd)/OOPS to /OOPS of container

root@acb8561561ca:/usr/local$ cd /jedi/code/comfv3

root@acb8561561ca:/jedi/code/comfv3$ cd release/v0/exp/

root@acb8561561ca:/jedi/code/comfv3/release/v0/exp$ ./build.sh macgnu

...
...
...
Elapsed time 66 seconds. Compiling HYDRO=Y 32BIT=N finished
+ cp /home/xinzhang/jedi/code/comfv3/release/v0/exp/../../../tests/fv3_1.exe ../NEMS/exe/fv3_gfs_hydro.
prod.64bit.x
+ rm /home/xinzhang/jedi/code/comfv3/release/v0/exp/../../../tests/fv3_1.exe
+ exit 0
```

13. Check out and build the CRTM V2.2.3 code

Checkout and Build CRTM V2.2.3

```
root@acb8561561ca:/usr/local$ cd /jedi/code

root@acb8561561ca:/jedi/code> git clone https://YourUserName@bitbucket.org/jcsda/crtm-release.git
crtm_v2.2.3 # please replace Your_UserName with your bitbucket username

root@acb8561561ca:/jedi/code> cd crtm_v2.2.3/

root@acb8561561ca:/jedi/code/crtm_v2.2.3> source config-setup/gfortran.setup

root@acb8561561ca:/jedi/code/crtm_v2.2.3>./configure --prefix=/jedi/build/crtm

root@acb8561561ca:/jedi/code/crtm_v2.2.3> make

root@acb8561561ca:/jedi/code/crtm_v2.2.3> make check

root@acb8561561ca:/jedi/code/crtm_v2.2.3> make install
```

What's inside the JEDI docker image?

When you start a Docker container instance from the jcsda/jedi-docker, we already prepapre the **gnu version 7.2 compilers** and most of the necessary libraiestools for GSI, OOPS, WRF etc., all libraries are installed under **/usr/local**, which include

- git
- git flow
- emacs
- open-mpi v2.1.0
- zlib v1.2.11
- szip v2.1.1
- jpeg v9b
- png v1.4.19
- jasper v1.900.2
- hdf5 v1.8.17
- freetype v2.5.5
- netcdf-c v4.4.11
- netcdf-fortran v4.4.4
- lapack v3.7.0
- parallel-netcdf v1.8.1
- xerces-c v3.1.4
- esmf v7.0.0
- udunites-2 v2.2.24
- nco v4.6.6
- grib_api v1.21.0
- cdo v1.8.2
- boost v1.65.1
- eigen3 v3.3.4
- pio 1.7.1
- ecbuild
- eckit
- fckit

The major NCEP libraries are also installed at :

- /nwprod/lib/bacio/v2.0.1/libbacio_v2.0.1_4.a
- /nwprod/lib/bacio/v2.0.1/libbacio_v2.0.1_8.a
- /nwprod/lib/ip/v2.0.0/libip_v2.0.0_4.a
- /nwprod/lib/ip/v2.0.0/libip_v2.0.0_8.a
- /nwprod/lib/ip/v2.0.0/libip_v2.0.0_d.a
- /nwprod/lib/sigio/v2.0.1/lib/libsigio_v2.0.1_4.a
- /nwprod/lib/sigio/v2.0.1/libsigio_v2.0.1_4.a
- /nwprod/lib/sp/v2.0.2/libsp_v2.0.2_4.a
- /nwprod/lib/sp/v2.0.2/libsp_v2.0.2_8.a
- /nwprod/lib/sp/v2.0.2/libsp_v2.0.2_d.a
- /nwprod/lib/w3emc/v2.2.0/libw3emc_v2.2.0_4.a
- /nwprod/lib/w3emc/v2.2.0/libw3emc_v2.2.0_8.a
- /nwprod/lib/w3emc/v2.2.0/libw3emc_v2.2.0_d.a
- /nwprod/lib/w3nco/v2.0.6/libw3nco_v2.0.6_4.a
- /nwprod/lib/w3nco/v2.0.6/libw3nco_v2.0.6_8.a
- /nwprod/lib/w3nco/v2.0.6/libw3nco_v2.0.6_d.a

Related articles

- Infrastructure Knowledge Base
- GNSSRO UFO Hackathon, August 21-27, 2018
- Running HOFX tests on the gnssro branch
- Build JEDI environment with Singularity
- Build JEDI environment with Docker