Building and Running the Containerized Global-Workflow

October 11, 2024

Mark Potts

Overview

Motivation

- Eliminates the need for users to build spack-stack
- Designed to provide a Tier-1-like interface for building software
- Greatly simplify the process of porting the Global-workflow to generic systems
- Allow non-NOAA researchers to work directly with UFS applications

Specifics

- Container based on spack-stack version 1.8.0
- Can be built from Docker images or downloaded directly from s3
- Provides an “externalize.sh” script that creates stand-alone wrappers that can be used on the host platform as if they were built natively
- This is a work in progress (not everything works)

Assumptions

- Some things are required on the host system
- Host has intelmpi or mpich installed (srun –mpi=pmi2 can work in place of mpiexec)
- Singularity or Apptainer is installed on your system
- Several hundred GB of free disk space
- At least 8 cores available for computation
- Preferred
  - slurm installed (rocoto_fake_slurm is an option)
  - rocoto installed (can be built in user space)
  - virtual python environment or ability to install python packages as needed

Getting the container

- Build it on your local machine using singularity/apptainer

```
singularity build --force ubuntu22.04-intel-ufs-env-v1.8.0.img \
docker://noaaepic/ubuntu22.04-intel-unified:v1.8.0
```

- Copy it from Tier-1 platform
  orion -- /work/noaa/epic-ps/role-epic-ps/containers
  hera -- /scratch1/NCEPDEV/nems/role.epic/containers
  jet -- /mnt/lfs4/HFIP/hfv3gfs/role.epic/containers
  noaacloud -- /contrib/EPIC/containers
  gaea – /gpfs/f5/epic/world-shared/containers
  derecho -- /glade/scratch/epicufsrt/containers

- Download it from s3

```
aws s3 cp --no-sign-request \
s3://noaa-ufs-gdas-pds/spack-stack-singularity-images/ubuntu22.04-intel-ufs-env-v1.8.0.img
```

Getting the data

- Download the fix files needed from aws

```
aws s3 sync --no-sign-request s3://noaa-nws-global-pds/fix fix
(these are huge, but not all are required)
```

- Download the ICSDIR data for C48 runs

```
mkdir -p ICSDIR/C48C48mx500/20240610
cd ICSDIR/C48C48mx500/20240610
aws s3 sync --no-sign-request
s3://noaa-nws-global-pds/data/ICSDIR/C48C48mx500/20240610/gfs.202
10323 gfs.20210323
```

Building the workflow

- Clone global-workflow

```
git clone --recursive https://github.com/noaa-emc/global-workflow
```

- download and untar patch

```
git clone
https://github.com/NOAA-EPIC/global-workflow-patch.git
cd global-workflow
tar xvfz ../global-workflow-patch/gw-patch.1.8.0d.tar.gz
```

- Shell into the container

```
export img=PATH-TO/ubuntu22.04-intel-ufs-env-v1.8.0.img
singularity shell -e -s /bin/mybash $img
```

- Set variables and run build script

```
export HOMEgfs=$PWD
export MACHINE_ID=container
source versions/build.container.ver
module use $PWD/modulefiles
module load module_base.container
cd sorc && ./build_all.sh -g -j 8
```

Prepare containerized executable to run on host system

- Run the link_workflow script in global-workflow/sorc
```
export FIX_DIR=PATH_TO_DOWNLOADED_FIX_FILES
./link_workflow.sh
```
- Move the global-workflow/exec directory
```
cd .. && mv exec container-exec
```
- Create externalized wrapper scripts for all G-W executables
```
/opt/container-scripts/externalize.sh -e $PWD/exec container-exec/*
```
- Externalize several other required scripts/executables
```
/opt/container-scripts/externalize.sh -e $PWD/exec/bin $WGRIB2
/opt/container-scripts/externalize.sh -e $PWD/exec
$prod_util_ROOT/bin/*
```
- Exit the container shell

Edit defaults and host particulars

- Open up /global-workflow/parm/config/gfs/yaml/defaults.yaml

```
FHMAX_GFS: 12
DO_TRACKER: “NO”
DO_GENESIS: “NO”
DO_METP: “NO”
```

- Open up global-workflow/workflow/hosts/container.yaml

```
DMPDIR: '/data2/${USER}'
HOMEDIR: '/home/${USER}'
STMP: '/data2/${USER}'
PTMP: '/data2/${USER}'
```

- Configure slurm for your site

```
SCHEDULER: slurm
ACCOUNT: ''
QUEUE: ''
QUEUE_SERVICE: ''
PARTITION_BATCH: ''
PARTITION_SERVICE: ''
RESERVATION: ''
CLUSTERS: ''
```

Set up experiment

- Set MACHINE_ID to “container”
  ```
  export MACHINE_ID=CONTAINER
  ```
- Load modules for intelmpi
- Set path to include exec
  ```
  export PATH=$PATH:/data2/sandbox/global-workflow/exec
  ```
- Set wgrib2_ROOT
  ```
  export wgrib2_ROOT=/data2/sandbox/global-workflow/exec
  ```
- Run setup_expt.py for forecast-only
  ```
  ./setup_expt.py gfs forecast-only --start cold --pslot c48_atm --app
  ATM --resdetatmos 48 --idate 2021032312 --edate 2021032312 --comroot
  /data2/comroot --icsdir=/data2/ICSDIR/C48C48mx500/20240610   --expdir
  /data2/expdir
  ```
- Run setup.xml script
  ```
  ./setup_xml.py /data2/expdir/c48_atm
  ```
- cd to expdir and start workflow with rocotorun
  ```
  rocotorun  -w c48_atm.xml -d c48_atm.db -v 10
  ```

Potential issues

- Extra python modules to install

```
pip install python-dateutil
pip install xarray
```

- No slurm?
  - Just copy and run the scripts produced directly
  - Use rocoto_fake_slurm from https://github.com/ufs-community/ufs-srweather-app
    - ufs-srweather-app/ufs/rocoto_fake_slurm
- Don’t need the full fix directories from s3.

```
aer -> fix/aer/20220805
chem -> fix/chem/20220805
cpl -> fix/cpl/20230526
gsi -> fix/gsi/20240208
mom6 -> fix/mom6/20240416
sfc_climo -> fix/sfc_climo/20220805
verif -> fix/verif/20220805
```

```
am -> fix/am/20220805
cice -> fix/cice/20240416
datm -> fix/datm/20220805
lut -> fix/lut/20220805
orog -> fix/orog/20231027
ugwd -> fix/ugwd/20240624
wave -> fix/wave/20240105
```

Useful rocoto commands

- Check overall status of workflow

  `rocotostat -w c48_atm.xml -d c48_atm.db -v 10`

```
       CYCLE                    TASK                       JOBID                STATE          EXIT STATUS      TRIES      DURATION
==========================================================================================================================
202103231200           gfs_stage_ic                     3729158            SUCCEEDED                    0          1           3.0
202103231200           gfs_fcst_seg0  druby://10.90.197.249:46381           SUBMITTING                    -          0           0.0
202103231200      gfs_atmos_prod_f000                           -                    -                    -          -           -
202103231200      gfs_atmos_prod_f003                           -                    -                    -          -           -
202103231200      gfs_atmos_prod_f006                           -                    -                    -          -           -
202103231200      gfs_atmos_prod_f009                           -                    -                    -          -           -
202103231200      gfs_atmos_prod_f012                           -                    -                    -          -           -
202103231200                 gfs_arch                           -                    -                    -          -           -
202103231200              gfs_cleanup                           -                    -                    -          -           -
mpotts@emc-lw-mpotts:rt/emc-lw-mpotts/mpotts/expdir/c48_atm 3$
```

- Rewind a step that failed

  `rocotorewind -w c48_atm.xml -d c48_atm.db -v 10 -c 202103231200 -t gfs_fcst_seg0`

- Check to see status details of a step

  `rocotorecheck -w c48_atm.xml -d c48_atm.db -v 10 -c 202103231200-t gfs_fcst_seg0`

- Force the completion of a step

  `rocotocomplete -w c48_atm.xml -d c48_atm.db -v 10 -c 202103231200-t gfs_fcst_seg0`