

FFV2 CONTAINER IMPLEMENTATION

Amalia Iriza-Burca (NMA)
Razvan Dobre

- Some **problems encountered** with implementing FFV2 on newer platforms / incompatibility issues.
- For the moment, **proposed solution**: **containers**

WHY:

- modern way of packaging HPC software
- HPC users can bring the local environment into the HPC platform faster
- less dependent on the HPC platform environment
- just the container runtime is needed

Summary of What we did

- created a Makefile that helps package the Met R library into a Linux Container
- the build process depends on Met R source files, Podman and Apptainer

Apptainer is the only container runtime required to run the container once is built (it has to be available on the HPC platform)

Creating the container

- Clone the repo from <https://github.com/dobrerazvan/met-container>
- **run `make sif`** - builds the container on your local machine. Alternatively you can download prebuilt containers from **TBD**

Using the container

Running R

apptainer exec <sif_file> R

Running demo files

apptainer exec <sif_file> scores_by_date|aggregate|scores_by_station <params>

Code

Issues 3

Pull requests 4

Actions

Projects

Security and quality

Insights

Settings

Files

master

Go to file

admin

cosmo

dace

lmn / cosmo

Add file

...

dobrerazvan Building containers for dace and ff cosmo tools (#94) 649978b · 1 hour ago History

Name	Last commit message	Last commit date
..		
dace	Building containers for dace and ff cosmo tools (#94)	1 hour ago
ff	Building containers for dace and ff cosmo tools (#94)	1 hour ago

Name	Last commit message	Last commit date
..		
Dockerfile	Building containers for dace and ff cosmo tools (#94)	2 hours ago
Makefile	Building containers for dace and ff cosmo tools (#94)	2 hours ago
README.md	Building containers for dace and ff cosmo tools (#94)	2 hours ago
run_ffv2.sh	Building containers for dace and ff cosmo tools (#94)	2 hours ago

README.md

✎ ☰

FFV2 container: portable image and on-demand use cases

- Build and run the FFV2 R package in a Rocky Linux 9 container.
- image built with Podman and exported as an ****Apptainer SIF**** for portability;
- run the SIF with ``apptainer run`` (or ``singularity run``).
- run specific use cases via a wrapper / override the installed package with modified R source for ad-hoc runs.

Build and test

```
``bash
make build  # build image r-rocky9-ffv2:<TAG> (Podman)
make test   # verify FFV2 loads
make sif    # build SIF r-rocky9-ffv2-<TAG>.sif for running with Apptainer
````
```

- Copy the ``.sif`` file to the machine(s) where you want to run;
- no container runtime other than **Apptainer** is required there.

## Apptainer/Singularity (SIF)

- On the build host: ``make sif`` (produces ``r-rocky9-ffv2-<TAG>.sif``).
- Copy the ``.sif`` file to the target machine. Run with: ``apptainer run r-rocky9-ffv2-<TAG>.sif ...`` (or ``singularity run ...``).

## OCI (Podman/Docker)

- On the build host: ``make export`` (writes ``r-rocky9-ffv2-<TAG>.tar``).
- Copy the ``.tar`` to the target, then: ``podman load -i r-rocky9-ffv2-<TAG>.tar`` (or ``docker load -i ...``).

## Check after copy

- Verify FFV2 loads with the SIF:

```
```bash
apptainer exec r-rocky9-ffv2-<TAG>.sif R -e "library(FFV2); cat('FFV2 loaded
successfully\n')"
```

Running a use case (on-demand)

The image entrypoint is a wrapper that runs one of three demo scripts by name. Bind-mount a ****workspace**** directory that contains your namelists (and, as per namelist, input/output paths). Use Apptainer's ``-B`` to bind host paths into the container.

Usage:

```
```bash
apptainer run -B /path/to/workspace:/workspace r-rocky9-ffv2-<TAG>.sif <use_case> <args...>
```
```

Use cases and arguments:

Use case

Arguments (after use case)

scores_by_date

namelist OBSTYPE FCST_TYPE no_cores

aggregate

namelist OBSTYPE FCST_TYPE no_cores

scores_by_station

namelist OBSTYPE FCST_TYPE PARAM_CATEG no_cores

- *basically, same as when running the original FFV2 scores*

Examples (replace `*<TAG>*` with your build tag, e.g. `20250310-1200`):

```
```bash
Scores by date (SYNOP DET, 6 cores)
apptainer run -B /host/workspace:/workspace r-rocky9-ffv2-<TAG>.sif scores_by_date
/workspace/namelist_verSYNOP.ni SYNOP DET 6

Aggregate
apptainer run -B /host/workspace:/workspace r-rocky9-ffv2-<TAG>.sif aggregate
/workspace/namelist_verSYNOP.ni SYNOP DET 6

Scores by station (CONT, 6 cores)
apptainer run -B /host/workspace:/workspace r-rocky9-ffv2-<TAG>.sif scores_by_station
/workspace/namelist.ni SYNOP DET CONT 6
```
```

See wrapper usage (& use-case names) without running a job:

```
```bash
apptainer run r-rocky9-ffv2-<TAG>.sif
```
```

Example Usage (operational)

```
apptainer run -e PATH/r-rocky9-ffv2-20260310-2108.sif option NAMELIST  
same_options_as_FFV2_scripts
```

```
apptainer run -e ${HOME}/CARMA/FFV2-main/run/r-rocky9-ffv2-20260310-2108.sif scores_by_date  
/export/home/amalia.iriza/CARMA/FFV2-main/namelists/SYNOP_27FEB2MAR26.ni SYNOP DET 6
```

```
apptainer run -e ${HOME}/CARMA/FFV2-main/run/r-rocky9-ffv2-20260310-2108.sif aggregate  
/export/home/amalia.iriza/CARMA/FFV2-main/namelists/SYNOP_27FEB2MAR26.ni SYNOP DET 6
```

```
apptainer run -e ${HOME}/CARMA/FFV2-main/run/r-rocky9-ffv2-20260310-2108.sif scores_by_station  
/export/home/amalia.iriza/CARMA/FFV2-main/namelists/SYNOP_27FEB2MAR26.ni SYNOP DET CONT 6
```

Odd runs with modified R source (R CMD INSTALL)

- use ****edited**** FFV2 code without rebuilding the image:
 1. *On the host, put the FFV2 ****source**** tree (e.g. `FFV2-main` with `DESCRIPTION`, `R/`, etc.) in a directory.*
 2. *Bind that directory at the fixed path ****`/mnt/FFV2`**** in the container.*
 3. *The wrapper will detect `/mnt/FFV2` and run ``R CMD INSTALL /mnt/FFV2`` before invoking the demo, so the modified package is used for that run.*

Example

```
```bash
apptainer run -B /host/workspace:/workspace -B /host/path/to/FFV2-main:/mnt/FFV2 \
 r-rocky9-ffv2-<TAG>.sif scores_by_date /workspace/namelist.nl SYNOP DET 6
```
```

Install manually and then run the wrapper (e.g. in an interactive shell):

```
``bash
apptainer shell -B /host/FFV2-main:/mnt/FFV2 -B /host/workspace:/workspace r-rocky9-ffv2-<TAG>.sif
# inside container:
R CMD INSTALL /mnt/FFV2
/usr/local/bin/run_ffv2.sh scores_by_date /workspace/namelist.nl SYNOP DET 6
``
```

Interactive R shell - get an R prompt instead of running the wrapper:

```
``bash
apptainer exec r-rocky9-ffv2-<TAG>.sif R
``
```

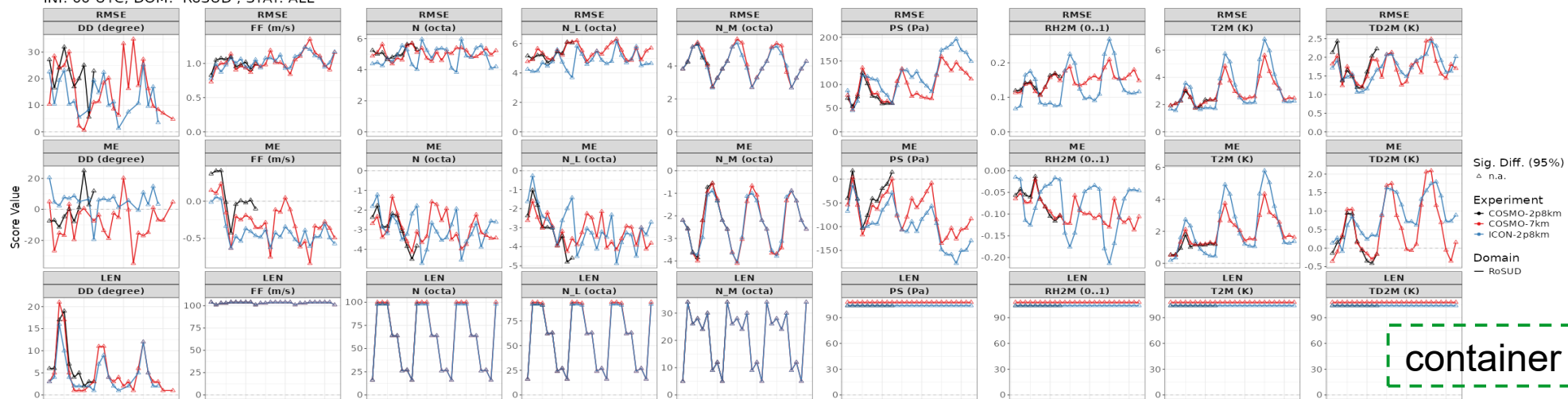
Makefile targets

| Target | Description |
|---------------|--|
| build | Build image with Podman (depends on Dockerfile, FFV2-main, run_ffv2.sh) |
| test | Run container to verify FFV2 loads |
| sif | Build Apptainer SIF `r-rocky9-ffv2-<TAG>.sif` (for `apptainer run` on target machines) |
| export | Write OCI archive `r-rocky9-ffv2-<TAG>.tar` for podman/docker load on another machine |

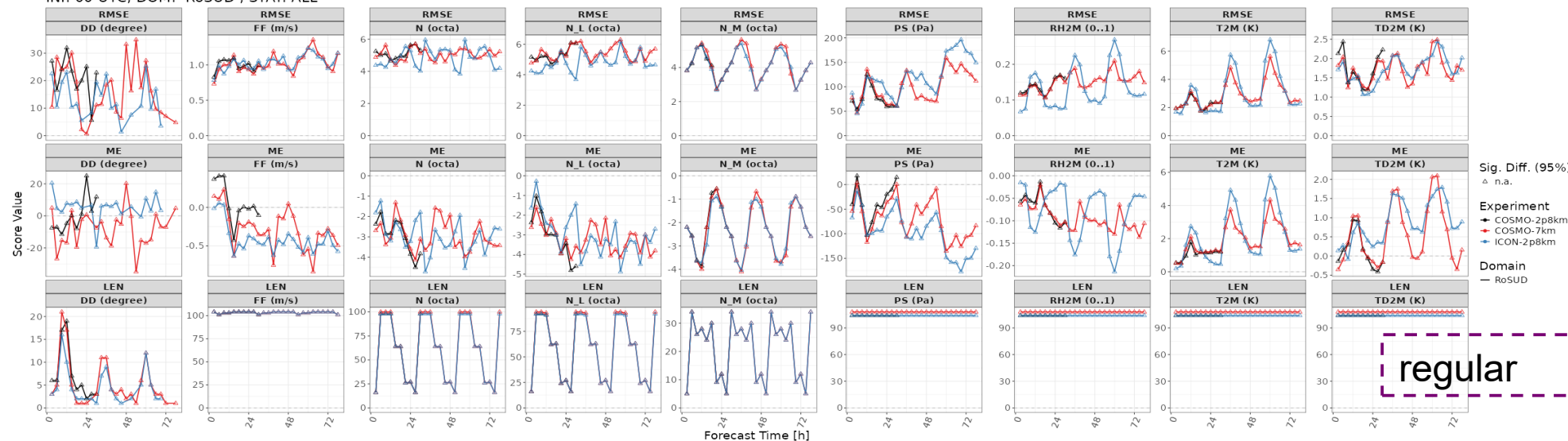
Results

• Identical to regular implementation

2026/02/27-00UTC - 2026/03/02-21UTC
INI: 00 UTC, DOM: RoSUD, STAT: ALL



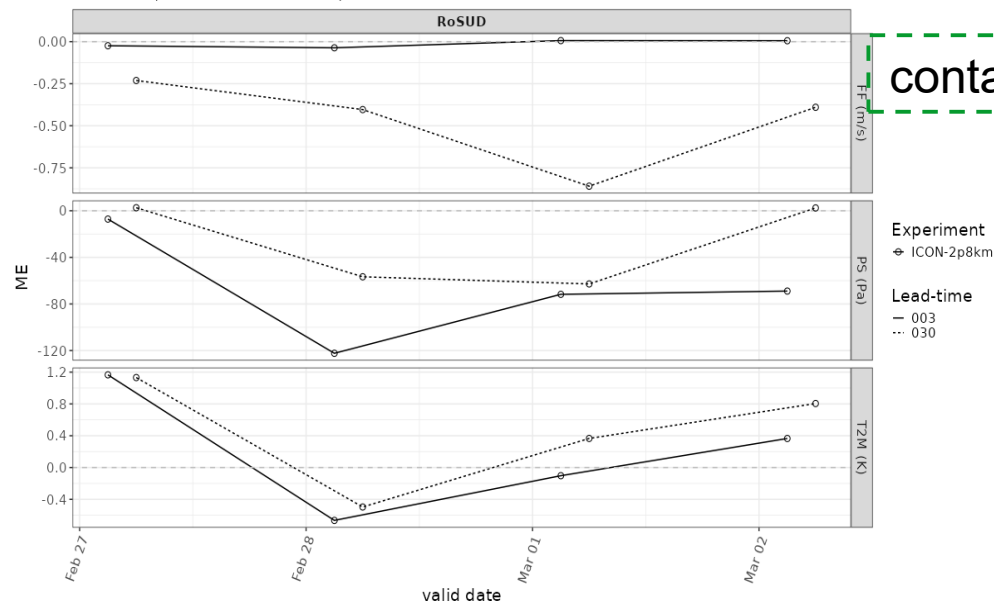
2026/02/27-00UTC - 2026/03/02-21UTC
INI: 00 UTC, DOM: RoSUD, STAT: ALL



Results

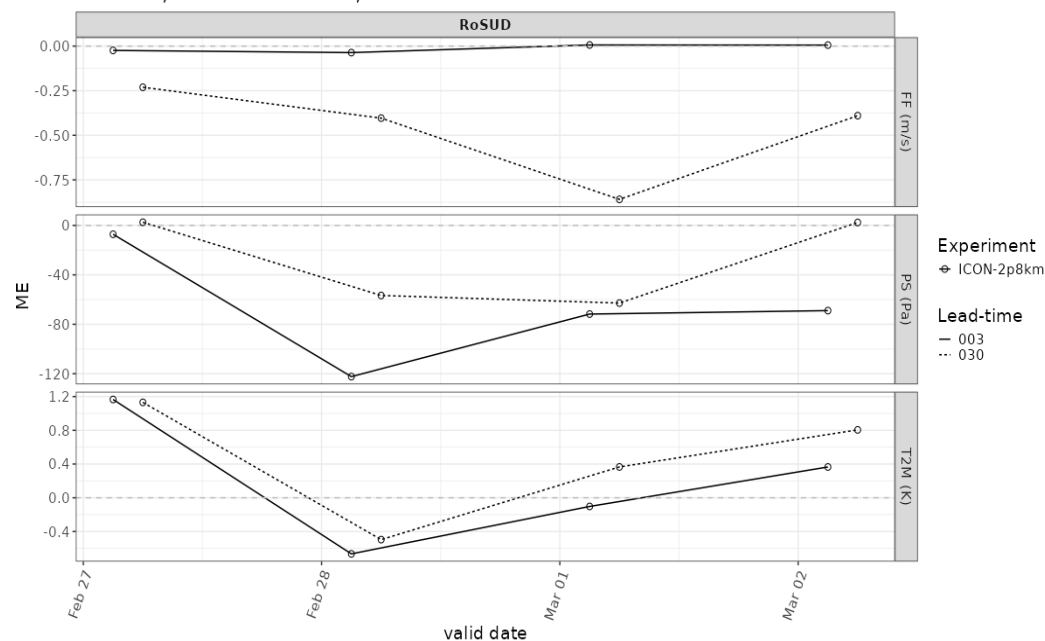
• Identical to regular implementation

2026.02.27-00UTC - 2026.03.02-21UTC
INI: 00 , DOMAIN: RoSUD , STATIONS: ALL



regular

2026.02.27-00UTC - 2026.03.02-21UTC
INI: 00 , DOMAIN: RoSUD , STATIONS: ALL



Summary of What we did

- created a Makefile that helps package the software into a Linux Container
- the build process depends on source files, Podman and Apptainer

Apptainer is the only container runtime required to run the container once is built (it has to be available on the HPC platform)

Creating the container

- Clone the repo from <https://github.com/dobrerazvan/met-container>
- **run `make sif`** - builds the container on your local machine. Alternatively you can download prebuilt containers from

Using the container

- It has yet to be parallelized and tested

Thank you!

Questions?