Cotainr on LUMI

Christian Schou Oxvig

LUMI User Support Team (LUST)

Danish e-infrastructure Consortium (DeiC)

DeiC

LUMI

September 27th, 2023

Motivation

- If you use Scientific Python packages, you probably also use conda/pip for managing your Python environments.
  - **$** conda env create --file my_conda_env.yml
- On LUMI, we tell you **NOT** to install a lot of Python packages via conda/pip on the Lustre filesystems – use a container instead!
  1. In general, you can't build containers on LUMI.
  2. Get a local Linux box/VM and install Singularity/Apptainer.
  3. Create a Singularity build recipe.
  4. Build your container locally.
  5. Copy your container to LUMI.
- Wouldn't it be nice to be able to "convert" your conda environment to a container on LUMI?
  - **$** cotainr build my_container.sif --system=lumi-g --conda-env=my_conda_env.yml

```
name:
lumi_cotainr_demo
channels:
  - conda-forge
dependencies:
  - python=3.11
  - numpy=1.25.2
  - pip
  - pip:
    - env-var==1.0.1
```

Demo

- Load cotainr module
  - `$ module load LUMI`
  - `$ module load cotainr`
- Build a container
  - `$ cotainr build my_container.sif --system=lumi-c --conda-env=my_conda_env.yml`
- Run the container like any other Singularity container, e.g.
  - `$ singularity shell my_container.sif`
  - `$ srun --account=<YOUR_PROJECT> --partition=debug singularity exec my_container.sif python3 my_script.py`

Benefits of using cotainr

- Manage your conda environments separately as containers
  - Enhanced reproducibility
  - No "entanglement" of Python installations for different projects/applications
  - No problem with hitting file qouta limits on LUMI
  - You play nicely with the Lustre file system
- Made with portability in mind
  - If your Python packages are built for a given "architecture", e.g. x86_64 or ROCm, you can copy your container to another HPC system having the same "architecture" and reuse it.

Documentation

- LUMI documentation
  - Cotainr - https://docs.lumi-supercomputer.eu/software/containers/singularity/#building-containers-using-the-cotainr-tool
  - Python packages install guidelines - https://docs.lumi-supercomputer.eu/software/installing/python/
- Cotainr documentation - https://cotainr.readthedocs.io/
- LUMI cotainr examples - https://github.com/DeiC-HPC/cotainr/tree/main/examples/LUMI

Even more to come

- Provide LUMI specific base images with the --system option:
  - LUMI-G: ROCm stack + aws-ofi-rccl plug-in (+ MIOpen kernels?)
  - LUMI-C: Cray MPICH compatible MPI
- Support other user space package managers:
  - R?
  - Maybe even Spack?
- <Your idea for a great use case for cotainr>

Feedback

- Cotainr is open-source and hosted on GitHub: https://github.com/DeiC-HPC/cotainr
  - **Bug reports**, **questions**, and **enhancement proposals** are very welcome on the cotainr GitHub issue tracker.
  - Pull requests are also very welcome, but please read the development section of the documentation before submitting a PR: https://cotainr.readthedocs.io/en/stable/development