Apptainer (Singularity)コンテナの利用

最終更新日: 2025/1/28

はじめに

Apptainer (Singularity)はコンテナ仮想化のプラットフォームで、Docker と良く似た立ち位置のソフトウェアです。Apptainer は HPC 環境での利用を想定して設計されているため、実行したユーザーの権限で実行できたり(root 権限や特殊な設定は不要)、カレントディレクトリが自動的に mount されてそのまま使えたり、GPU の利用が容易であったりと、スパコンのような環境で使いやすくなっています。

情報は随時追加していきます。

コンテナイメージ

Docker のイメージを利用することも可能です。NVIDIA NGC にある docker イメージなども利用可能です。以下のコマンドを実行すると NGC にあるイメージから apptainer (singularity) 用のイメージを構築できます。

```
$ apptainer pull docker://nvcr.io/nvidia/pytorch:23.11-py3
```

以下のようなサイトから既存のイメージを取得できます。

- Docker Hub: https://hub.docker.com/
- NVIDIA NGC: https://catalog.ngc.nvidia.com/containers

コンテナイメージの作成について

ccfep にてコンテナイメージを作成することができます。以下のようなファイル(ubuntu24_04.def とします)から apptainer イメージを作成する例を示します。ubuntu 24.04 環境の /opt に miniforge を導入し、その base 環境にいくつかパッケージを導入しています。コンテナインスタンスの起動時にはその conda 環境を読み込む設定(%environment)になっています。

```
Bootstrap: docker
From: ubuntu:24.04

%post
  apt-get -y update
  apt-get -y upgrade
  apt-get -y install \
    build-essential \
    wget \
    bzip2 \
    git
  apt-get -y clean
  wget -c https://github.com/conda-forge/miniforge/releases/latest/download/Miniforge3-Linux-x86_64.sh
  /bin/sh Miniforge3-Linux-x86_64.sh -bfp /opt/miniforge
  /opt/miniforge/bin/conda shell.bash hook > /opt/miniforge/conda_init.sh
  . /opt/miniforge/conda_init.sh
  conda install opencv numpy scipy scikit-learn jax
  conda install pytest pandas sphinx curl glib glob2 isort pango
  conda install path pathlib2 pathtools psutil cmake
  conda install jupyterlab eigen boost transformers boost-cpp
  conda install h5py markdown matplotlib
  conda install scikit-image sqlite jupyter
  # show list to review
  conda list

%environment
  . /opt/miniforge/conda_init.sh

%labels
  Author RCCS
  Version 0.0.1

%help
  Sample python environment for RCCS supercomputer system.
```

```
%runscript

```

コンテナイメージの作成(ユーザー権限で作成すれば自動的に --fakeroot が付加されます)。

```
$ apptainer build ubuntu24_04.sif ubuntu24_04.def
```

コンテナインスタンス内で shell を起動する

```
$ apptainer shell ubuntu24_04.sif
(base) python
Python 3.10.14 | packaged by conda-forge | (main, Mar 20 2024, 12:45:18) [GCC 12.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

作成されたコンテナインスタンス中でコマンドを実行

```
$ apptainer exec ubuntu24_04.sif gcc -dumpfullversion
13.2.0
$ apptainer exec ubuntu24_04.sif head -3 /etc/os-release
PRETTY_NAME="Ubuntu Noble Numbat (development branch)"
NAME="Ubuntu"
VERSION_ID="24.04"
```

注意点

- 上記と同等のサンプルを /apl/apptainer/sample-ubuntu24.04/ 以下にも置いています。
- GPU を使う場合、ドライバはコンテナではなくホスト側のものが利用される点にご注意下さい。
  - apptainer コマンドに --nv オプションを追加する必要もあります。apptainer run --nv (sif ファイル名) のような形で実行します。
  - コンテナがホストのものより新しい GPU ドライバのバージョンを要求した場合、エラーとなります。
  - 事前にコンテナの要求するドライバのバージョン、もしくは CUDA のバージョン(RCCS で最新のもの以下のバージョンならばおそらく大丈夫です)をご確認下さい。
- MPI でのノード内並列についてはコンテナイメージに MPI 環境を導入し、それを利用すれば問題無くできます。
  - こちらの場合はコンテナインスタンス内で mpirun を実行してしまって大丈夫です。
- MPI でのノード間並列についてはコンテナ外に依存する部分が発生するため容易ではありませんが、一応は可能です。
  - mpirun で singularity を複数立ち上げるような形になります。
  - コンテナ内とコンテナ外の MPI 環境を合わせる必要が発生します。
  - 仮想環境を複数立ち上げるために性能が出にくくなる可能性があります。
  - 避けられるのであれば避けた方が無難です。
- ホームディレクトリは自動的に mount されるため、bind オプションで指定する必要はありません。
- /apl 以下に導入されたソフトも同時に使うような場合には --bind に /apl:/apl を追加するようお願いします。
  - apptainer run --bind /apl:/apl --nv (sif ファイル名) のように指定します。
  - 場合によっては /gwork なども追加すると便利かもしれません。