Containers have great advantage for application portability in different environments by isolating process resources and namespaces with a small performance overhead, thus it has been rapidly getting popular in a wide range of science fields.

However, there are problems in container image configuration when run in multiple HPC environments, and it requires users to have knowledge of systems, container runtimes, image format and compatibility of those used in HPC environments.

In this study, we introduce our HPC container workflow on multiple supercomputing environments that have different system/library specifications (ABCI, TSUBAME3.0). Our workflow provides custom container image configurations for HPC environments taking into account differences in container runtime, container image, and library compatibility between the host and inside of the container. We also show the parallel performance of our application in each HPC environment.

**HPC Container Workflow with HPCCM**

**Advantages of our container workflow**

1. Single python recipe (HPCCM) to generate container recipe
   - for Dockerfile, Singularity definition

2. No image format conversion
   - avoiding image compatibility problems

3. Configuration for specific MPI library
   - specifying version of libraries with user arguments

- > hpccm --recipe megadock.py
- --format singularity
- --userarg fttw=3.3.8 ompi=2.1.2 ope=True

**Code Availability**

- MEGADOCK-HPCCM recipe
  - Dockerfile & Singularity definition for ABCI and TSUBAME 3.0
  - Dataset & experiment information

- MEGADOCK project repository
  - Software manuals (install, run, etc.)
  - Dockerfile for GPU enabled environment