# ROCm-Containers Documentation

***Release latest***

**Jun 22, 2022**

# FREQUENTLY ASKED QUESTIONS

This is the user guide for ROCm Container modules deployed in Purdue High Performance Computing clusters. More information about our center is avaiable here (https://www.rcac.purdue.edu).

If you have any question, contact me(Yucheng Zhang) at: zhan4429@purdue.edu

CHAPTER

ONE

# FREQUENTLY ASKED QUESTIONS

Question

- Answer

Question

- Answer

Question

- Answer

Question

- Answer

CHAPTER

TWO

# CP2K

## 2.1 Introduction

CP2K is a quantum chemistry and solid state physics software package that can perform atomistic simulations of solid state, liquid, molecular, periodic, material, crystal, and biological systems. CP2K provides a general framework for different modeling methods such as DFT using the mixed Gaussian and plane waves approaches GPW and GAPW. Supported theory levels include DFTB, LDA, GGA, MP2, RPA, semi-empirical methods (AM1, PM3, PM6, RM1, MNDO, ...), and classical force fields (AMBER, CHARMM, ...). CP2K can do simulations of molecular dynamics, metadynamics, Monte Carlo, Ehrenfest dynamics, vibrational analysis, core level spectroscopy, energy minimization, and transition state optimization using NEB or dimer method. CP2K is written in Fortran 2008 and can be run efficiently in parallel using a combination of multi-threading, MPI, and HIP/CUDA. For more information, please check: Home page: http://www.cp2k.org/ Docker: https://www.amd.com/en/technologies/infinity-hub/cp2k

## 2.2 Versions

- 20210311–h87ec1599

## 2.3 Commands

- cp2k.psmp
- cp2k.popt
- cp2k_shell.psmp
- dumpdcd.psmp
- graph.psmp
- grid_miniapp.psmp
- xyz2dcd.psmp
- benchmark
- mpirun
- mpiexec
- ompi_info

## 2.4 Module

You can load the modules by:

```
module load rocmcontainers
module load cp2k
```

## 2.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run cp2k on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=cp2k
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers cp2k
```

CHAPTER

THREE

# DEEPSPEED

## 3.1 Introduction

DeepSpeed is a deep learning optimization library that makes distributed training easy, efficient, and effective. DeepSpeed delivers extreme-scale model training for everyone, from data scientists training on massive supercomputers to those training on low-end clusters or even on a single GPU. For more information, please check: Home page: https://www.deepspeed.ai Docker: docker://rocm/deepspeed

## 3.2 Versions

- rocm4.2_ubuntu18.04_py3.6_pytorch_1.8.1

## 3.3 Commands

- deepspeed
- python
- python3
- python3.6
- ipython
- ipython3
- convert-caffe2-to-onnx
- convert-onnx-to-caffe2
- estimator_ckpt_converter
- import_pb_to_tensorboard
- tensorboard
- tflite_convert
- mpirun
- mpiexec
- ompi_info

## 3.4 Module

You can load the modules by:

```
module load rocmcontainers
module load deepspeed
```

## 3.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run deepspeed on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=deepspeed
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers deepspeed
```

CHAPTER

FOUR

# GROMACS

## 4.1 Introduction

GROMACS is a molecular dynamics application designed to simulate Newtonian equations of motion for systems with hundreds to millions of particles. GROMACS is designed to simulate biochemical molecules like proteins, lipids, and nucleic acids that have a lot of complicated bonded interactions. This container, based on a released version of GROMACS, is an AMD beta version with ongoing optimizations. This container only supports up to a 4 GPU configuration. For more information, please check: Home page: https://www.gromacs.org Docker: https://www.amd.com/en/technologies/infinity-hub/gromacs

## 4.2 Versions

- 2020.3

## 4.3 Commands

- gmx
- gmx_mpi
- demux.pl
- xplor2gmx.pl
- mpirun
- mpiexec
- ompi_info

## 4.4 Module

You can load the modules by:

```
module load rocmcontainers
module load gromacs
```

## 4.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run gromacs on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=gromacs
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers gromacs
```

CHAPTER

FIVE

# NAMD

## 5.1 Introduction

NAMD is a parallel molecular dynamics code designed for high-performance simulation of large biomolecular systems. NAMD uses the popular molecular graphics program VMD for simulation setup and trajectory analysis, but is also file-compatible with AMBER, CHARMM, and X-PLOR. For more information, please check: Home page: http://www.ks.uiuc.edu/Research/namd/ Docker: https://www.amd.com/en/technologies/infinity-hub/namd

## 5.2 Versions

- 2.15a2

## 5.3 Commands

- charmrun
- flipbinpdb
- flipdcd
- namd2
- psfgen
- sortreplicas

## 5.4 Module

You can load the modules by:

```
module load rocmcontainers
module load namd
```

## 5.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run namd on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=namd
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers namd
```

CHAPTER

SIX

# OPENMM

## 6.1 Introduction

OpenMM is a high-performance toolkit for molecular simulation. It can be used as an application, a library, or a flexible programming environment. OpenMM includes extensive language bindings for Python, C, C++, and even Fortran. The code is open source and developed on GitHub, licensed under MIT and LGPL. This module defines program installation directory (note: inside the container!) as environment variable $OPENMM_PATH. Once again, this is not a host path, this path is only available from inside the container. Most likely you will not need it for production simulations, but it might be occasionally needed for benchmarks or access to container innards. With the way this module is organized, you should be able to use this variable freely with containerized commands like python3 $OPENMM_PATH/examples/benchmarks.py –help For more information, please check: Home page: https://openmm.org Docker: https://www.amd.com/en/technologies/infinity-hub/openmm

## 6.2 Versions

- 7.4.2

## 6.3 Commands

- python
- python3
- python3.8
- python2
- python2.7
- run-benchmarks

## 6.4 Module

You can load the modules by:

```
module load rocmcontainers
module load openmm
```

## 6.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run openmm on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=openmm
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers openmm
```

CHAPTER

SEVEN

# PYTORCH

## 7.1 Introduction

PyTorch is an optimized tensor library for deep learning using GPUs and CPUs. For more information, please check: Home page: https://pytorch.org/ Docker: docker://rocm/pytorch

## 7.2 Versions

- 1.8.1-rocm4.2-ubuntu18.04-py3.6
- 1.9.0-rocm4.2-ubuntu18.04-py3.6

## 7.3 Commands

- python
- python3
- python3.6
- convert-caffe2-to-onnx
- convert-onnx-to-caffe2
- mpirun
- mpiexec
- ompi_info

## 7.4 Module

You can load the modules by:

```
module load rocmcontainers
module load pytorch
```

## 7.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run pytorch on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=pytorch
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers pytorch
```

CHAPTER

EIGHT

# SPECFEM3D

## 8.1 Introduction

SPECFEM3D Cartesian simulates acoustic (fluid), elastic (solid), coupled acoustic/elastic, poroelastic or seismic wave propagation in any type of conforming mesh of hexahedra (structured or not.) It can, for instance, model seismic waves propagating in sedimentary basins or any other regional geological model following earthquakes. It can also be used for non-destructive testing or for ocean acoustics. This module conflicts with RCAC 'openmpi' modules - unload them before use (there is a built-in OpenMPI inside the container). For more information, please check: Home page: https://geodynamics.org/cig/software/specfem3d/ Docker: https://www.amd.com/en/technologies/infinity-hub/specfem3d

## 8.2 Versions

- 20201122–h9c0626d1

## 8.3 Commands

- xadd_model_iso
- xcheck_mesh_quality
- xclip_sem
- xcombine_sem
- xcombine_surf_data
- xcombine_vol_data
- xcombine_vol_data_vtk
- xconvert_skewness_to_angle
- xconvolve_source_timefunction
- xcreate_movie_shakemap_AVS_DX_GMT
- xdecompose_mesh
- xdecompose_mesh_mpi
- xdetect_duplicates_stations_file
- xgenerate_databases
- xinverse_problem_for_model

- xmeshfem3D
- xmodel_update
- xsmooth_sem
- xspecfem3D
- xsum_kernels
- xsum_preconditioned_kernels
- benchmark
- mpirun
- mpiexec
- ompi_info

## 8.4 Module

You can load the modules by:

```
module load rocmcontainers
module load specfem3d
```

## 8.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run specfem3d on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=specfem3d
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers specfem3d
```

CHAPTER

NINE

# SPECFEM3D_GLOBE

## 9.1 Introduction

SPECFEM3D Globe simulates global and regional (continental-scale) seismic wave propagation. This module conflicts with RCAC 'openmpi' modules - unload them before use (there is a built-in OpenMPI inside the container). For more information, please check: Home page: https://geodynamics.org/cig/software/specfem3d_globe/ Docker: https://www.amd.com/en/technologies/infinity-hub/specfem3d_globe

## 9.2 Versions

- 20210322–h1ee10977

## 9.3 Commands

- xadd_model_iso
- xadd_model_tiso
- xadd_model_tiso_cg
- xadd_model_tiso_iso
- xaddition_sem
- xclip_sem
- xcombine_AVS_DX
- xcombine_paraview_strain_data
- xcombine_sem
- xcombine_surf_data
- xcombine_vol_data
- xcombine_vol_data_vtk
- xconvolve_source_timefunction
- xcreate_cross_section
- xcreate_header_file
- xcreate_movie_AVS_DX

- xcreate_movie_GMT_global
- xdetect_duplicates_stations_file
- xdifference_sem
- xextract_database
- xinterpolate_model
- xmeshfem3D
- xsmooth_laplacian_sem
- xsmooth_sem
- xspecfem3D
- xsum_kernels
- xsum_preconditioned_kernels
- xwrite_profile
- benchmark
- mpirun
- mpiexec
- ompi_info

## 9.4 Module

You can load the modules by:

```
module load rocmcontainers
module load specfem3d_globe
```

## 9.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run specfem3d_globe on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=specfem3d_globe
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out
```

(continues on next page)

(continued from previous page)

```
module --force purge
ml rocmcontainers specfem3d_globe
```

CHAPTER

# TEN

# TENSORFLOW

## 10.1 Introduction

TensorFlow is an end-to-end open source platform for machine learning. For more information, please check: Home page: https://www.tensorflow.org Docker: docker://rocm/tensorflow

## 10.2 Versions

- 2.5-rocm4.2-dev

## 10.3 Commands

- python
- python3
- python3.6
- ipython
- ipython3
- bazel
- estimator_ckpt_converter
- horovodrun
- import_pb_to_tensorboard
- jupyter
- saved_model_cli
- tensorboard
- tflite_convert
- mpirun
- mpiexec
- ompi_info

## 10.4 Module

You can load the modules by:

```
module load rocmcontainers
module load tensorflow
```

## 10.5 Example job

**Warning:** Using `#!/bin/sh -l` as shebang in the slurm job script will cause the failure of some biocontainer modules. Please use `#!/bin/bash` instead.

To run tensorflow on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=tensorflow
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml rocmcontainers tensorflow
```