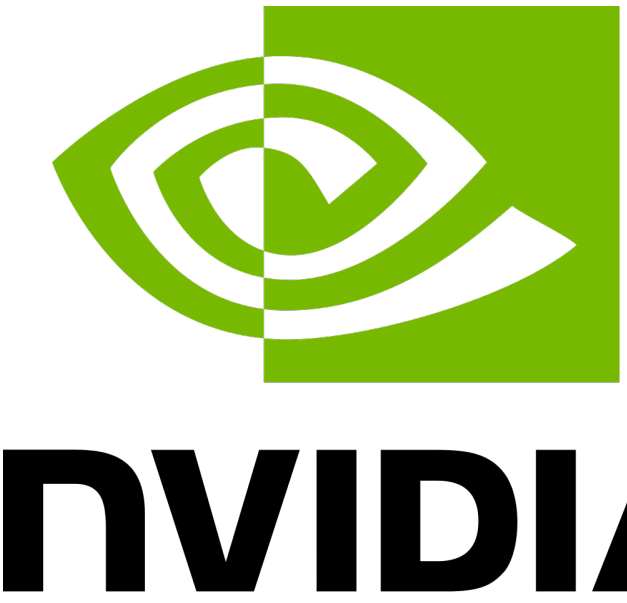
NGC Documentation

Release latest

Jul 20, 2022

FREQUENTLY ASKED QUESTIONS



This is the user guide for NGC 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

AUTODOCK

2.1 Introduction

The AutoDock Suite is a growing collection of methods for computational docking and virtual screening, for use in structure-based drug discovery and exploration of the basic mechanisms of biomolecular structure and function. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:autodock

2.2 Versions

- 2020.06

2.3 Commands

- autodock_gpu_128wi

2.4 Module

You can load the modules by:

```
module load ngc
module load autodock
```

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 autodock on our clusters:

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

module --force purge
ml ngc autodock
```

CHAPTER

THREE

CHROMA

3.1 Introduction

The Chroma package provides a toolbox and executables to carry out calculation of lattice Quantum Chromodynamics (LQCD). It is built on top of the QDP++ (QCD Data Parallel Layer) which provides an abstract data parallel view of the lattice and provides lattice wide types and expressions, using expression templates, to allow straightforward encoding of LQCD equations. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:chroma

3.2 Versions

- 2018-cuda9.0-ubuntu16.04-volta-openmpi
- 2020.06
- 2021.04

3.3 Commands

- chroma
- hmc
- mpirun

3.4 Module

You can load the modules by:

```
module load ngc
module load chroma
```

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 chroma on our clusters:

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

module --force purge
ml ngc chroma
```

CHAPTER

FOUR

GAMESS

4.1 Introduction

The General Atomic and Molecular Electronic Structure Systems (GAMESS) program simulates molecular quantum chemistry, allowing users to calculate various molecular properties and dynamics. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:gamess

4.2 Versions

- 17.09-r2-libcchem

4.3 Commands

- rungms

4.4 Module

You can load the modules by:

```
module load ngc
module load gamess
```

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 gamess on our clusters:

```
#!/bin/bash
#SBATCH -A myallocation     # Allocation name
#SBATCH -t 1:00:00
```

(continues on next page)

(continued from previous page)

```
#SBATCH -N 1
#SBATCH -n 1
#SBATCH --job-name=gamess
#SBATCH --mail-type=FAIL,BEGIN,END
#SBATCH --error=%x-%J-%u.err
#SBATCH --output=%x-%J-%u.out

module --force purge
ml ngc gamess
```

CHAPTER

FIVE

GROMACS

5.1 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. More info on GROMACS can be found at http://www.gromacs.org/ For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:gromacs

5.2 Versions

- 2018.2
- 2020.2
- 2021.3
- 2021

5.3 Commands

- gmx

5.4 Module

You can load the modules by:

```
module load ngc
module load gromacs
```

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 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 ngc gromacs
```

CHAPTER

SIX

JULIA

6.1 Introduction

The Julia programming language is a flexible dynamic language, appropriate for scientific and numerical computing, with performance comparable to traditional statically-typed languages. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:julia

6.2 Versions

- v1.5.0
- v2.4.2

6.3 Commands

- julia

6.4 Module

You can load the modules by:

```
module load ngc
module load julia
```

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 julia on our clusters:

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

module --force purge
ml ngc julia
```

CHAPTER

SEVEN

LAMMPS

7.1 Introduction

Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) is a software application designed for molecular dynamics simulations. It has potentials for solid-state materials (metals, semiconductor), soft matter (biomolecules, polymers) and coarse-grained or mesoscopic systems. It can be used to model atoms or, more generically, as a parallel particle simulator at the atomic, meso, or continuum scale. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:lammps

7.2 Versions

- 10Feb2021
- 15Jun2020
- 24Oct2018
- 29Oct2020

7.3 Commands

- lmp
- mpirun

7.4 Module

You can load the modules by:

```
module load ngc
module load lammps
```

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 lammps on our clusters:

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

module --force purge
ml ngc lammps
```

CHAPTER

EIGHT

MILC

8.1 Introduction

MILC represents part of a set of codes written by the MIMD Lattice Computation (MILC) collaboration used to study quantum chromodynamics (QCD), the theory of the strong interactions of subatomic physics. It performs simulations of four dimensional SU(3) lattice gauge theory on MIMD parallel machines. "Strong interactions" are responsible for binding quarks into protons and neutrons and holding them all together in the atomic nucleus. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:milc

8.2 Versions

- quda0.8-patch4Oct2017

8.3 Commands

- mpirun
- su3_rhmd_hisq

8.4 Module

You can load the modules by:

```
module load ngc
module load milc
```

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 milc on our clusters:

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

module --force purge
ml ngc milc
```

CHAPTER

NINE

NAMD

9.1 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: NGC: https://ngc.nvidia.com/catalog/containers/hpc:namd

9.2 Versions

- 2.13-multinode
- 2.13-singlenode
- 3.0-alpha3-singlenode

9.3 Commands

- charmrun
- flipbinpdb
- flipdcd
- namd3
- psfgen
- sortreplicas
- vmd

9.4 Module

You can load the modules by:

```
module load ngc
module load namd
```

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 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 ngc namd
```

CHAPTER

TEN

NVHPC

10.1 Introduction

The NVIDIA HPC SDK is a comprehensive suite of compilers, libraries and tools essential to maximizing developer productivity and the performance and portability of HPC applications. The NVIDIA HPC SDK C, C++, and Fortran compilers support GPU acceleration of HPC modeling and simulation applications with standard C++ and Fortran, OpenACC directives, and CUDA. GPU-accelerated math libraries maximize performance on common HPC algorithms, and optimized communications libraries enable standards-based multi-GPU and scalable systems programming. Performance profiling and debugging tools simplify porting and optimization of HPC applications, and containerization tools enable easy deployment on-premises or in the cloud. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/nvidia:nvhpc

10.2 Versions

- 20.11
- 20.7
- 20.9
- 21.5
- 21.9

10.3 Commands

- nvc
- nvc++
- nvfortran
- nvcc
- pgcc
- pgc++
- pgfortran
- cuda-gdb
- ncu

- nv-nsight-cu-cli
- nvprof
- nsight-sys
- nsys

10.4 Module

You can load the modules by:

```
module load ngc
module load nvhpc
```

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 nvhpc on our clusters:

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

module --force purge
ml ngc nvhpc
```

CHAPTER

ELEVEN

PARAVIEW

11.1 Introduction

ParaView is an open-source, multi-platform data analysis and visualization application. This ParaView container is enabled with the NVIDIA IndeX plugin and the OptiX ray-tracing backend. It can be used in tandem with an official ParaView]] .. ver .. [[client or standalone as ParaView Web. Note: no ParaView client GUI in this container. However, ParaView Web Visualizer app is included for a ParaView-like experience inside a web browser. You can start ParaView Web with a 'pvweb' command and point your browser to proper http://HOST:PORT/ Default port is 8080 ('–port NNNN' to change, '–help' for help). For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/nvidia-hpcvis:paraview

11.2 Versions

- 5.9.0

11.3 Commands

- pvserver
- pvbatch
- pvpython
- pvdataserver
- pvrenderserver
- mpirun

11.4 Module

You can load the modules by:

```
module load ngc
module load paraview
```

11.5 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 paraview on our clusters:

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

module --force purge
ml ngc paraview
```

CHAPTER

TWELVE

PYTORCH

12.1 Introduction

PyTorch is a GPU accelerated tensor computational framework with a Python front end. Functionality can be easily extended with common Python libraries such as NumPy, SciPy, and Cython. Automatic differentiation is done with a tape-based system at both a functional and neural network layer level. This functionality brings a high level of flexibility and speed as a deep learning framework and provides accelerated NumPy-like functionality. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/nvidia:pytorch

12.2 Versions

- 20.02-py3
- 20.03-py3
- 20.06-py3
- 20.11-py3
- 20.12-py3
- 21.06-py3
- 21.09-py3

12.3 Commands

- python
- python3

12.4 Module

You can load the modules by:

```
module load ngc
module load pytorch
```

12.5 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 ngc pytorch
```

CHAPTER

THIRTEEN

QMCPACK

13.1 Introduction

QMCPACK is an open-source, high-performance electronic structure code that implements numerous Quantum Monte Carlo algorithms. Its main applications are electronic structure calculations of molecular, periodic 2D and periodic 3D solid-state systems. Variational Monte Carlo (VMC), diffusion Monte Carlo (DMC) and a number of other advanced QMC algorithms are implemented. By directly solving the Schrodinger equation, QMC methods offer greater accuracy than methods such as density functional theory, but at a trade-off of much greater computational expense. Distinct from many other correlated many-body methods, QMC methods are readily applicable to both bulk (periodic) and isolated molecular systems. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:qmcpack

13.2 Versions

- v3.5.0

13.3 Commands

- mpirun
- qmcpack

13.4 Module

You can load the modules by:

```
module load ngc
module load qmcpack
```

13.5 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 qmcpack on our clusters:

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

module --force purge
ml ngc qmcpack
```

CHAPTER

FOURTEEN

QUANTUM_ESPRESSO

14.1 Introduction

Quantum ESPRESSO is an integrated suite of Open-Source computer codes for electronic-structure calculations and materials modeling at the nanoscale based on density-functional theory, plane waves, and pseudopotentials. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:quantum_espresso

14.2 Versions

- v6.6a1
- v6.7

14.3 Commands

- mpirun
- pw.x

14.4 Module

You can load the modules by:

```
module load ngc
module load quantum_espresso
```

14.5 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 quantum_espresso on our clusters:

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

module --force purge
ml ngc quantum_espresso
```

CHAPTER

FIFTEEN

RAPIDSAI

15.1 Introduction

The RAPIDS suite of software libraries gives you the freedom to execute end-to-end data science and analytics pipelines entirely on GPUs. It relies on NVIDIA® CUDA® primitives for low-level compute optimization, but exposes that GPU parallelism and high-bandwidth memory speed through user-friendly Python interfaces. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/nvidia:rapidsai:rapidsai

15.2 Versions

- 0.12
- 0.13
- 0.14
- 0.15
- 0.16
- 0.17
- 21.06
- 21.10

15.3 Commands

- ipython3
- ipython3
- jupyter
- python
- python3
- python3.8

15.4 Module

You can load the modules by:

```
module load ngc
module load rapidsai
```

15.5 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 rapidsai on our clusters:

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

module --force purge
ml ngc rapidsai
```

CHAPTER

SIXTEEN

RELION

16.1 Introduction

RELION (for REgularized LIkelihood OptimizatioN) implements an empirical Bayesian approach for analysis of electron cryo-microscopy (Cryo-EM). Specifically it provides methods of refinement of singular or multiple 3D reconstructions as well as 2D class averages. RELION is an important tool in the study of living cells. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:relion

16.2 Versions

- 2.1.b1
- 3.0.8
- 3.1.0
- 3.1.2
- 3.1.3

16.3 Commands

- mpirun
- relion
- relion_refine_mpi

16.4 Module

You can load the modules by:

```
module load ngc
module load relion
```

16.5 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 relion on our clusters:

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

module --force purge
ml ngc relion
```

CHAPTER

SEVENTEEN

TENSORFLOW

17.1 Introduction

TensorFlow is an open-source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) that flow between them. This flexible architecture lets you deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device without rewriting code. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/nvidia:tensorflow

17.2 Versions

- 20.02-tf1-py3
- 20.02-tf2-py3
- 20.03-tf1-py3
- 20.03-tf2-py3
- 20.06-tf1-py3
- 20.06-tf2-py3
- 20.11-tf1-py3
- 20.11-tf2-py3
- 20.12-tf1-py3
- 20.12-tf2-py3
- 21.06-tf1-py3
- 21.06-tf2-py3
- 21.09-tf1-py3
- 21.09-tf2-py3

17.3 Commands

- python
- python3

17.4 Module

You can load the modules by:

```
module load ngc
module load tensorflow
```

17.5 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 ngc tensorflow
```

CHAPTER

EIGHTEEN

TORCHANI

18.1 Introduction

TorchANI is a PyTorch implementation of ANI(Accurate NeurAl networK engINe for Molecular Energies), created and maintained by the Roitberg group. TorchANI contains classes like AEVComputer, ANIModel, and EnergyShifter that can be pipelined to compute molecular energies from the 3D coordinates of molecules. It also include tools to: deal with ANI datasets(e.g. ANI-1, ANI-1x, ANI-1ccx, ANI-2x) at torchani.data, import various file formats of NeuroChem at torchani.neurochem, and more at torchani.utils. For more information, please check: NGC: https://ngc.nvidia.com/catalog/containers/hpc:torchani

18.2 Versions

- 2021.04

18.3 Commands

- mpirun
- python
- python3
- jupyter

18.4 Module

You can load the modules by:

```
module load ngc
module load torchani
```

18.5 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 torchani on our clusters:

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

module --force purge
ml ngc torchani
```