李政道研究所

TSUNG-DAO LEE INSTITUTE

BHAC Code Hands-on

Yosuke Mizuno (水野 陽介)

Tsung-Dao Lee Institute,
Shanghai Jiao Tong University

BHAC tutorial II, TDLI, October 13th, 2022

Before start

For using your computer

- BHAC use Fortran90 and MPI. Need install before start
- For Mac/Linux: please installed gfortran (intel fortran) and openmpi (mpich)
- For Windows: recommend to use virtual box (virtual machine) to install Linux system (like Ubuntu). Then inside linux system, install gfortran and openmpi
- Better to install python3 (+vtk) for data analysis and visualisation
- For quick visualisation of 2D/3D data, it is better to use Visit or Paraview

For using TDLI Astro cluster

- intel and gfortran compilers are install. Please load parallel_studio_xe/2020 (intel)

BHAC hangs-on 1

Getting Started

1. Getting the Code

BHAC website: https://bhac.science/

- Install the code to the ~/codes/bhac directory. To download the code, the following commands should do the trick:

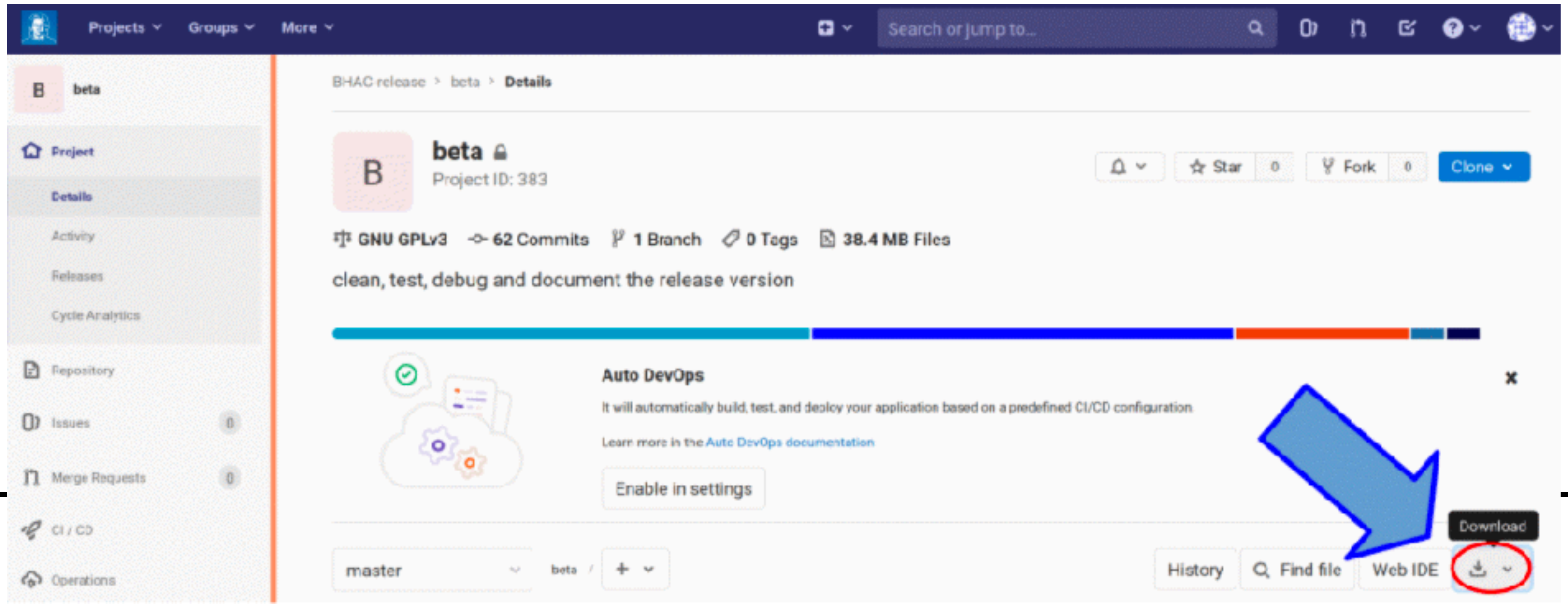
```
$ cd ~
$ make codes
$ cd codes
```

- Download BHAC code

1) **Using git:**

```
$ git clone https://gitlab.itp.uni-frankfurt.de/BHAC-release/bhac.git
```

2) **Downloading a compressed folder:**



BHAC release › beta › Details

beta

Project ID: 383

GNU GPLv3 · 62 Commits · 1 Branch · 0 Tags · 38.4 MB Files

clean, test, debug and document the release version

Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration

Learn more in the Auto DevOps documentation

Getting Started

2.Installation

Provide BHAC_DIR environment variable holding the path to the code

```
$ echo ‘export BHAC_DIR=$HOME/codes/bhac' >> ~/.bashrc (adjust your installed directory)
```

or manually adding following entry into .bashrc (or .bash_profile)

```
export BHAC_DIR = $HOME/codes/bhac (adjust your installed directory)
```

To use scripts more conveniently, the following line in ~/.bashrc (or bash_profile) is recommended although not essential:

```
PATH=“$BHAC_DIR:$BHAC_DIR/tools:./:$PATH”
```

```
$ source ~/.bashrc
```

if you need, ($ source ~/.bash_profile)

Getting Started

For TDLI Astro cluster

Software in cluster is handled by “module”

check available module

```
$ module avail
```

load module

```
$ module load parallel_studio_xe/2020
```

Check your loaded module

```
$ module list
```

Running Test Problem

- Running 1D shock tube test problem.
- To run the code, a good practice is to create a new directory outside $BHAC_DIR.
- Create a directory bhac_runs in the home directory and copy the problem set-up

```
$ mkdir ~/bhac_runs
$ cd ~/bhac_runs
$ cp -r $BHAC_DIR/tests/rmhd/shockTube ./
$ cd shockTube
```

Running Test Problem

- You will see some *.t file in your directory.

amrvacusr.t: the whole problem setup for initial conditions and boundary conditions

amrvacusrpar.t: used to provide additional global variables for your setup

- Before compile, we have several machine specific definition of compilers in $BHAC_DIR/arch
- If you want to use intel compiler

```
$ BHAC_DIR/setup.pl -arch=default
```

- If you want to use gfortran compiler

```
$ BHAC_DIR/setup.pl -arch=gfortran
```

- Note: For TDLI Astro cluster usage, need to change compiler option

Running Test Problem

- For TDLI Astro cluster, need to change compiler option. First go to inside arch directory

```
$ cd $BHAC_DIR/arch
```

- Edit “default.defs” file (emacs or vim)

Old

```
F90=mpif90
FFLAGS = -c
F90FLAGS = -xHost -O3 -FR -implicitnone
LINK= $(F90)  $(F90FLAGS)
```

New

```
F90=mpiifort
FFLAGS = -c
F90FLAGS = -fc=ifort -xHost -O3 -FR -implicitnone
LINK= $(F90)  $(F90FLAGS)
```

- After edit file, return to working directory

```
$ BHAC_DIR/setup.pl -arch=default
```

or editing “makefile”, ARCH = default.defs

Running Test Problem

- make compile

```
$ make
```

- after successful compilation, create a output directory

```
$ mkdir output
```

- To run the code,

```
$ mpiexec -n 4 ./bhac -i amrvac1D.par > output/out
```

- This will run BHAC on 4 cores of your machine using pre-defined parameter-file amrvac1D.par (-i).

Running Test Problem

- Several new files will appear inside the directory ‘output’, ending in .dat .vtu and one file called amrvac.log – this is the log file updated during computation.
- .dat files are used for restarts
- .vtu files contain output data to be visualised in Paraview (Visit) (or python)

Visualising the output

1. Setting up the BHAC python tools

- Install ipython3 and python3-vtk7
- Need to set-up path to BHAC python tools in .bashrc file
  - export PYTHONPATH = $BHAC_DIR/tools/python:$PYTHONPATH
  - $ source ~/.bashrc

Visualising the output

2. Plotting Data

- Inside the shockTube directory, run ipython3 with

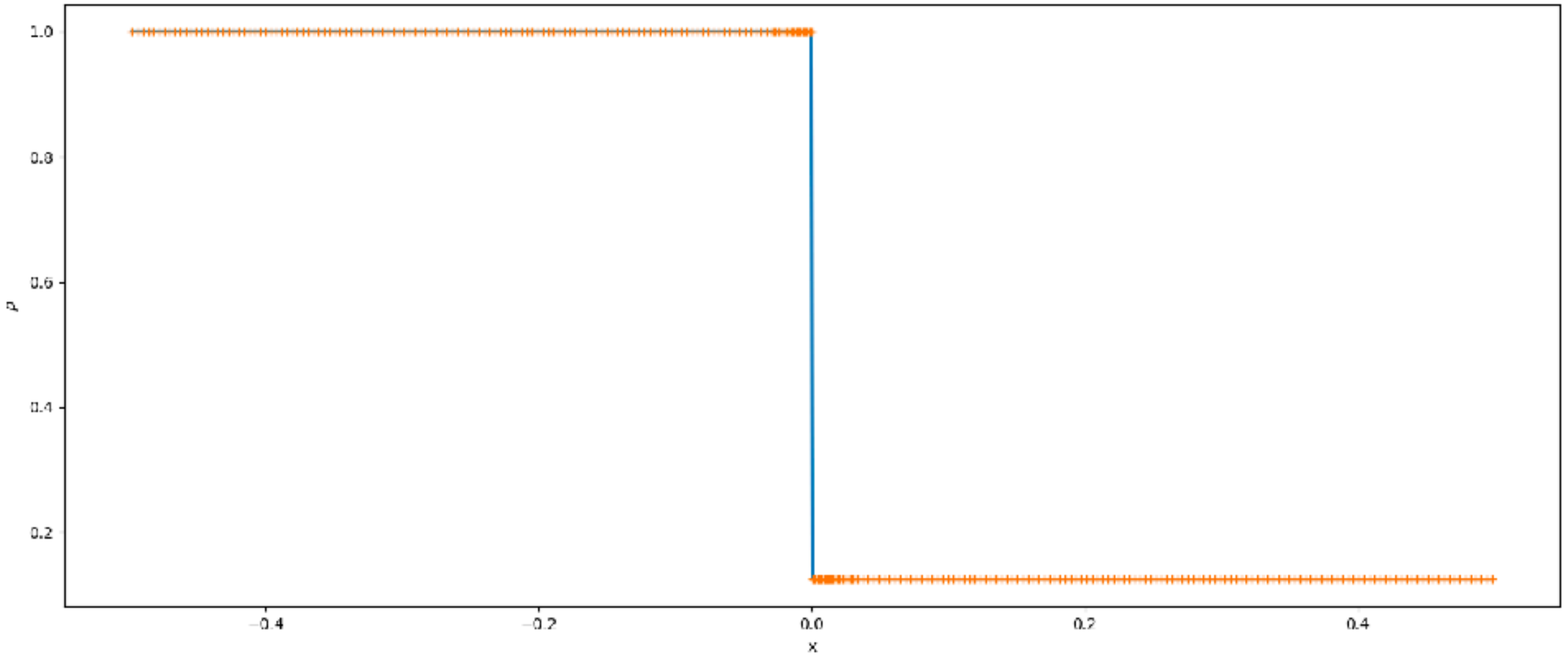
  -
    ```
    $ ipython3 —pylab
    ```
- load BHAC python tools

  -
    ```
    $ import read, amrplot
    ```
- load one of the output flies

  -
    ```
    $ d=read.load(0,file=‘output/data1D’,type=‘vtu’)
    ```

Visualising the output

- Now data is stored on the object “d”and we can query several quantities by looking at its attributes.
  - $ **d.time** simulation time
  - $ **d.rho** density array
  - $ x=d.getCenterPoints()
  - $ plt.plot(x,d.rho,’+’)



- Next read second snapshot
  - $ e=read.load(1,file=‘output/data1D’,type=‘vtu’)
- and repeat the steps. How change in different time? How does other quantities looks like?

BHAC hangs-on 2

Simulating Magnetized Bondi accretion

- Running 2D magnetized Bondi test problem.
- Follow first tutorial session, go to 'bhac_runs' directory,
  - $ cd ~/bhac_runs
- and copy 'Bondi2D' test from BHAC files
  - $ cp -r $BHAC_DIR/tests/rmhd/bondi2D ./
- Inside your bondi2D directory, compile the test (need to adjust compiler option in makefile)
  - $ make
- and create output directory using
  - $ mkdir output

Simulating Magnetized Bondi accretion

- Now running the simulation
  - $ mpiexec -n 1 ./bhac
- If you want to use multiple cores
  - $ mpiexec -n 4 ./bhac

Job script for TDLI Astro cluster

- For TDLI Astro cluster user, need to submit job script
- Create “job.sh”

```
#!/bin/bash
#SBATCH --partition=normal (or student)
#SBATCH --ntasks-per-node=4
#SBATCH --nodes=1
#SBATCH --time=1-00:00:00
#SBATCH --job-name=Bondi2D
#SBATCH --mail-type=ALL
#SBATCH -o output/out # STDOUT
#SBATCH -e output/err # STDERR

module load parallel_studio_xe/2020

mpirun -genv I_MPI_OFI_PROVIDER=tcp -np 4 ./bhac -i amrvac.par
```

- Then submit job: $sbatch job.sh

Job script for TDLI Astro cluster

- For TDLI Astro cluster user,
- Running job is managed by “slurm”
- Some useful command
- check available nodes: $sinfo
- check submitted job status: $squeue
- submit job script: $sbatch <job_script>
- cancel job: $scancel <job_ID>

viewing the output

- $ ipython3 —pylab
- $ import read, amrplot
- $ d= read.load(0,file=’output / data’, type=’vtu’)
- $ p1 = amrplot.polyplot (d.rho,d)
- Then you can see spherically symmetric density profile

BHAC hangs-on 3

Simulating Magnetized Torus

- Running 2D magnetized Fishbone-Moncrief torus test problem.
- Follow first tutorial session, go to ‘bhac_runs’ directory,
  - `$ cd ~/bhac_runs`
- and copy ‘FMtorus’ test from BHAC files
  - `$ cp -r $BHAC_DIR/tests/rmhd/FMtorus ./`
- Inside your FMtorus directory, make compile (need to adjust compiler option in makefile)
  - `$ make`
  - [Note]: in astro cluster, you will see the error. Need to change “amrvacusr.t” file.
- and create output directory using
  - `$ mkdir output`

Simulating Magnetized Bondi accretion

- Now running the simulation
  - `$ mpiexec -n 1 ./bhac`
- If you want to use multiple cores
  - `$ mpiexec -n 4 ./bhac`

- But 1 core simulations take too much time. So it is better to use cluster or reduce number of grid.

Job script for TDLI Astro cluster

- For TDLI Astro cluster user, need to submit job script
- Create “job.sh”

```
#!/bin/bash
#SBATCH --partition=normal (or student)
#SBATCH --ntasks-per-node=48
#SBATCH --nodes=1
#SBATCH --time=1-00:00:00
#SBATCH --job-name=2DFMtorus
#SBATCH --mail-type=ALL
#SBATCH -o output/out # STDOUT
#SBATCH -e output/err # STDERR

module load parallel_studio_xe/2020

mpirun -genv I_MPI_OFI_PROVIDER=tcp -np 48 ./bhac -i amrvac.par
```

- Then submit job: $sbatch job.sh