HPC Sample Job: Gromacs

Overview

Gromacs can be run using four main classes of resources:

- small jobs using CPU cores in a single computer,
- small jobs using CPU cores and 1-2 GPUs in a single computer,
- large jobs using CPU cores in multiple computers connected over a network, and
- large jobs using CPU cores and 1-2 GPUs per computer in multiple computers connected over a network.

Input files for all the examples below are from the HECBioSim project's benchmark suite.

Refer to the Slurm Quick Start User Guide for more information on Slurm scripts.

Single-computer, non-GPU Gromacs Job

Working from the HECBioSim 20k atom model, create a Slurm job script named `gromacs_multicore.sh` to run the job:

gromacs_multicore.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --cpus-per-task=28

INPUT=bench.tpr
OUTPUT=bench.log
module load gromacs
gmx mdrun -nt ${SLURM_CPUS_PER_TASK} -s ${INPUT} -g ${OUTPUT}
```

Submit the job from the login node with the command `sbatch gromacs_multicore.sh`, and when the job completes, you should have several new files and `bench.log`, containing the Gromacs output.

Final directory listing for Gromacs multicore job Expand source

Single-computer, GPU-enabled Gromacs Job

Working from the HECBioSim 20k atom model, edit the file `_bench.mdp` to adding a line ensuring that Gromacs uses the `verlet` cutoff scheme:

Section of Gromacs input file

```
cutoff-scheme           = verlet
```

and changing the line:

Section of Gromacs input file

```
rcoulomb                = 1.4
```

to:

Section of Gromacs input file

```
rcoulomb                = 1.2
```

to match the `rvdw` setting.

Regenerate the Gromacs .tpr file from the HECBioSim topology file and your updated molecular dynamics parameters file:

```
gmx grompp -f _bench.mdp -c bench.tpr -p 3NIR.top -o bench_gpu.tpr
```

Create a Slurm job script named `gromacs_gpu.sh` to run the job (the `--cpus-per-task` flag has been adjusted from 28 to 16 as one solution to how Gromacs will divide up the job):

- Training & Tutorials
- System Status
- Cyber Threat Bulletin
- Who Is Information Technology Services?
- Chat with a myTECH Support Representative
- Contact Us
- Frequently Asked Questions
- Report A Problem
- Apply for Student Work with ITS
- ITS Kiosk Landing Page
- COVID-19 ITS Response
- Ask Awesome-ChatBot Test Environment

gromacs_gpu.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --cpus-per-task=16
#SBATCH --gres=gpu:2

INPUT=bench_gpu.tpr
OUTPUT=bench_gpu.log
module load cuda80/toolkit gromacs
gmx mdrun -nt ${SLURM_CPUS_PER_TASK} -s ${INPUT} -g ${OUTPUT}
```

Submit the job from the login node with the command `sbatch gromacs_gpu.sh`, and when the job completes, you should have several new files and `bench_gpu.log`, containing the Gromacs output.

Final directory listing for Gromacs GPU job Expand source

Multi-computer, non-GPU Gromacs Job

Working from the HECBioSim 20k atom model, create a Slurm job script named `gromacs_mpi.sh` to run the job:

gromacs_mpi.sh

```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=2
#SBATCH --cpus-per-task=4

INPUT=bench_gpu.tpr
OUTPUT=bench_mpi.log
module load gromacs
mpirun `which mdrun_mpi` -ntomp ${SLURM_CPUS_PER_TASK} -s ${INPUT} -g ${OUTPUT}
```

Submit the job from the login node with the command `sbatch gromacs_mpi.sh`, and when the job completes, you should have several new files and `bench_mpi.log`, containing the Gromacs output.

Final directory listing for Gromacs MPI job Expand source

Multi-computer, GPU-enabled Gromacs Job

Working from the HECBioSim 20k atom model, edit the file `_bench.mdp` to adding a line ensuring that Gromacs uses the `verlet` cutoff scheme:

Section of Gromacs input file

```
cutoff-scheme           = verlet
```

and changing the line:

Section of Gromacs input file

```
rcoulomb                = 1.4
```

to:

Section of Gromacs input file

```
rcoulomb                = 1.2
```

to match the `rvdw` setting.

Regenerate the Gromacs .tpr file from the HECBioSim topology file and your updated molecular dynamics parameters file:

```
gmx grompp -f _bench.mdp -c bench.tpr -p 3NIR.top -o bench_mpi_gpu.tpr
```

Create a Slurm job script named `gromacs_mpi_gpu.sh` to run the job (the `--cpus-per-task` flag has been adjusted from 28 to 8 as one solution to how Gromacs will divide up the job):

gromacs_mpi_gpu.sh

```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=2
#SBATCH --cpus-per-task=8
```

```
#SBATCH --gres=gpu:2

INPUT=bench_gpu.tpr
OUTPUT=bench_mpi_gpu.log
module load cuda80/toolkit gromacs
mpirun `which mdrun_mpi` -ntomp ${SLURM_CPUS_PER_TASK} -s ${INPUT} -g ${OUTPUT}
```

Submit the job from the login node with the command `sbatch gromacs_mpi_gpu.sh`, and when the job completes, you should have several new files and `bench_mpi_gpu.log`, containing the Gromacs output.

Final directory listing for Gromacs MPI GPU job Expand source

Improving Performance for Gromacs Jobs

Please read the Gromacs documentation section Getting good performance from mdrun for more information about benchmarking Gromacs and improving your job runtimes.

Though it's assumed that adding more processors to a given job will automatically increase performance, various model sizes will reach peak performance with different processor counts. Additionally, simply adding minimal GPU support to an input file might not improve performance without more extensive changes.

How helpful was this information?

Your Rating: Results: 86 rates