Application Guide

V1.0 04/19/2023

List of Applications

1. **GROMACS**
2. **NAMD3**
3. **LAMMPS**
4. **CP2K**
5. **RELION v4**

Introduction

Login to Plexus with your registered user account.

Guide for GROMACS Application

Select GROMACS Docker Application.

Navigate to the Application Configuration screen.

Copy the following code block in the Run Script text box to run ADH_DODEC benchmark on an 8-GPU compute node.

```
nstlist=150; ntomp=8; cd /benchmarks/adh_dodec; tar -xvf adh_dodec.tar.gz;
gmx mdrun -pin on -nsteps 100000 -resetstep 90000 -ntmpi 8 -ntomp $ntomp -noconfout -nb gpu
-bonded gpu -pme gpu -npme 1 -v -nstlist $nstlist -gpu_id 01234567 -s topol.tpr
```

Copy the following code block in the Run Script text to run CELLULOSE_NVE benchmark on an 8-GPU compute node.

```
nstlist=200; ntomp=8; cd /benchmarks/cellulose_nve; tar -xvf cellulose_nve.tar.gz;
gmx mdrun -pin on -nsteps 100000 -resetstep 90000 -ntmpi 8 -ntomp $ntomp -noconfout -nb gpu
-bonded gpu -pme gpu -npme 1 -v -nstlist $nstlist -gpu_id 01234567 -s topol.tpr
```

Copy the following code block in the Run Script text box to run STMV benchmark on an 8-GPU compute node.

```
nstlist=400; ntomp=8; cd /benchmarks/stmv; tar -xvf stmv.tar.gz;
gmx mdrun -pin on -nsteps 100000 -resetstep 90000 -ntmpi 8 -ntomp $ntomp -noconfout -nb gpu
-bonded gpu -pme gpu -npme 1 -v -nstlist $nstlist -gpu_id 01234567 -s topol.tpr
```

How to get the Figure of Merit (FOM)

From the completed Workloads list, select the completed GROMACS job. Select the STDERR tab to view the stderr logs from the completed job. Scroll down to view the Performance output from the job. For example, here's an example output from STMV run showing **87.078 ns/day** as the FOM from that run.

```
-> Performance: 87.078 0.276
```

Guide for NAMD3 Application

Select NAMD3 Docker Application.

Navigate to the `Application Configuration` screen.

Copy the following code block in the `Run Script` text box to run `apoa1` benchmark concurrently on all GPUs on an 8-GPU compute node.

```
cd /examples
BENCHMARK=/examples/apoa1/apoa1.namd
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 0 --CUDASOAintegrate on +devices
0 2>&1 > gpu0 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 8 --CUDASOAintegrate on +devices
1 2>&1 > gpu1 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 16 --CUDASOAintegrate on +devices
2 2>&1 > gpu2 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 32 --CUDASOAintegrate on +devices
3 2>&1 > gpu3 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 64 --CUDASOAintegrate on +devices
4 2>&1 > gpu4 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 72 --CUDASOAintegrate on +devices
5 2>&1 > gpu5 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 80 --CUDASOAintegrate on +devices
6 2>&1 > gpu6 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 88 --CUDASOAintegrate on +devices
7 2>&1 > gpu7
wait
./ns_per_day.py gpu0
./ns_per_day.py gpu1
./ns_per_day.py gpu2
./ns_per_day.py gpu3
./ns_per_day.py gpu4
./ns_per_day.py gpu5
./ns_per_day.py gpu6
./ns_per_day.py gpu7
```

How to compute Figure of Merit (FOM)

From the completed Workloads list, select your completed NAMD3 job. Select the `STDOUT` tab to view `stdout` logs. The FOM for NAMD3 8-GPU workload is the sum of the eight `Nanoseconds per day` numbers. Here is an example output from `apoa1` benchmark run. The computed FOM for this example is **`862.694 Nanoseconds per day`** `(8 * 107.9 approx.)`.

```
Nanoseconds per day: 107.972
Mean time per step: 0.00160041
Standard deviation: 1.9379e-05
Nanoseconds per day: 107.949
Mean time per step: 0.00160076
Standard deviation: 1.59461e-05
Nanoseconds per day: 107.228
Mean time per step: 0.00161151
Standard deviation: 1.59286e-05
Nanoseconds per day: 107.572
Mean time per step: 0.00160637
Standard deviation: 6.90183e-06
Nanoseconds per day: 107.163
Mean time per step: 0.00161249
Standard deviation: 1.22992e-05
Nanoseconds per day: 108.366
Mean time per step: 0.0015946
Standard deviation: 1.39513e-05
Nanoseconds per day: 108.35
Mean time per step: 0.00159483
Standard deviation: 1.61872e-05
Nanoseconds per day: 108.094
Mean time per step: 0.00159861
Standard deviation: 1.68244e-05
```

Copy the following code block in the `Run Script` text to run `jac` benchmark concurrently on all GPUs on an 8-GPU compute node.

```
cd /examples
BENCHMARK=/examples/jac/jac.namd
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 0 --CUDASOAintegrate on +devices
0 2>&1 > gpu0 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 8 --CUDASOAintegrate on +devices
1 2>&1 > gpu1 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 16 --CUDASOAintegrate on +devices
2 2>&1 > gpu2 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 32 --CUDASOAintegrate on +devices
3 2>&1 > gpu3 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 64 --CUDASOAintegrate on +devices
4 2>&1 > gpu4 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 72 --CUDASOAintegrate on +devices
5 2>&1 > gpu5 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 80 --CUDASOAintegrate on +devices
6 2>&1 > gpu6 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 88 --CUDASOAintegrate on +devices
7 2>&1 > gpu7
wait
./ns_per_day.py gpu0
./ns_per_day.py gpu1
./ns_per_day.py gpu2
./ns_per_day.py gpu3
./ns_per_day.py gpu4
./ns_per_day.py gpu5
./ns_per_day.py gpu6
./ns_per_day.py gpu7
```

How to compute Figure of Merit (FOM)

From the completed Workloads list, select your completed NAMD3 job. Select the `STDOUT` tab to view `stdout` logs. The FOM for NAMD3 8-GPU workload is the sum of the eight `Nanoseconds per day` numbers.

Copy the following code block in the `Run Script` text box to run `f1atpase` benchmark concurrently on all GPUs on an 8-GPU compute node.

```
cd /examples
BENCHMARK=/examples/f1atpase/f1atpase.namd
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 0 --CUDASOAintegrate on +devices
0 2>&1 > gpu0 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 8 --CUDASOAintegrate on +devices
1 2>&1 > gpu1 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 16 --CUDASOAintegrate on +devices
2 2>&1 > gpu2 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 32 --CUDASOAintegrate on +devices
3 2>&1 > gpu3 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 64 --CUDASOAintegrate on +devices
4 2>&1 > gpu4 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 72 --CUDASOAintegrate on +devices
5 2>&1 > gpu5 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 80 --CUDASOAintegrate on +devices
6 2>&1 > gpu6 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 88 --CUDASOAintegrate on +devices
7 2>&1 > gpu7
wait
./ns_per_day.py gpu0
./ns_per_day.py gpu1
./ns_per_day.py gpu2
./ns_per_day.py gpu3
./ns_per_day.py gpu4
./ns_per_day.py gpu5
./ns_per_day.py gpu6
./ns_per_day.py gpu7
```

How to compute Figure of Merit (FOM)

From the completed Workloads list, select your completed NAMD3 job. Select the `STDOUT` tab to view `stdout` logs. The FOM for NAMD3 8-GPU workload is the sum of the eight `Nanoseconds per day` numbers.

Copy the following code block in the `Run Script` text box to run `stmv` benchmark concurrently on all GPUs on an 8-GPU compute node.

```
cd /examples
BENCHMARK=/examples/stmv/stmv.namd
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 0 --CUDASOAintegrate on +devices
0 2>&1 > gpu0 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 8 --CUDASOAintegrate on +devices
1 2>&1 > gpu1 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 16 --CUDASOAintegrate on +devices
2 2>&1 > gpu2 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 32 --CUDASOAintegrate on +devices
3 2>&1 > gpu3 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 64 --CUDASOAintegrate on +devices
4 2>&1 > gpu4 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 72 --CUDASOAintegrate on +devices
5 2>&1 > gpu5 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 80 --CUDASOAintegrate on +devices
6 2>&1 > gpu6 &
/opt/namd/bin/namd3 $BENCHMARK +p1 +pemap 88 --CUDASOAintegrate on +devices
7 2>&1 > gpu7
wait
./ns_per_day.py gpu0
./ns_per_day.py gpu1
./ns_per_day.py gpu2
./ns_per_day.py gpu3
./ns_per_day.py gpu4
./ns_per_day.py gpu5
./ns_per_day.py gpu6
./ns_per_day.py gpu7
```

How to compute Figure of Merit (FOM)

From the completed Workloads list, select your completed NAMD3 job. Select the `STDOUT` tab to view `stdout` logs. The FOM for NAMD3 8-GPU workload is the sum of the eight `Nanoseconds per day` numbers.

Guide for LAMMPS Application

Select the `LAMMPS` Docker Application

Navigate to the `Application Configuration` Screen

Copy the following code block in the `Run Script` text box to run `lj` benchmark.

```
SIZE=14; STEPS=600; GPUS=8; BENCHMARK="in.lj"; cd /benchmark;
/opt/lammps/tpl/openmpi/bin/mpirun -np $GPUS /opt/lammps/bin/lmp -k on g $GPUS -sf kk -pk
kokkos cuda/aware on neigh full comm device neigh/qeq full newton on -v x $SIZE -v y $SIZE
-v z $SIZE -v steps $STEPS -in $BENCHMARK -nocite
```

Copy the following code block in the `Run Script` text box to run eam benchmark.

```
SIZE=17; STEPS=300; GPUS=8; BENCHMARK="in.eam"; cd /benchmark;
/opt/lammps/tpl/openmpi/bin/mpirun -np $GPUS /opt/lammps/bin/lmp -k on g $GPUS -sf kk -pk
kokkos cuda/aware on neigh full comm device neigh/qeq full newton on -v x $SIZE -v y $SIZE
-v z $SIZE -v steps $STEPS -in $BENCHMARK -nocite
```

Copy the following code block in the `Run Script` text box to run `tersoff` benchmark.

```
SIZE=9; STEPS=300; GPUS=8; BENCHMARK="in.tersoff";cd /benchmark;
/opt/lammps/tpl/openmpi/bin/mpirun -np $GPUS /opt/lammps/bin/lmp -k on g $GPUS -sf kk -pk
kokkos cuda/aware on neigh half binsize 5.6 comm device neigh/qeq full newton on -v x
$SIZE -v y $SIZE -v z $SIZE -v steps $STEPS -in $BENCHMARK -nocite
```

Copy the following code block in the `Run Script` text box to run `reaxff` benchmark.

```
SIZE=48; STEPS=300; GPUS=8; BENCHMARK="in.reaxc.hns"; cd /benchmark;
/opt/lammps/tpl/openmpi/bin/mpirun -np $GPUS /opt/lammps/bin/lmp -k on g $GPUS -sf kk -pk
kokkos cuda/aware on neigh half comm device neigh/qeq full newton on -v x $SIZE -v y $SIZE
-v z $SIZE -v steps $STEPS -in $BENCHMARK -nocite
```

Copy the following code block in the `Run Script` text to run `snap` benchmark.

```
SIZE=23; STEPS=400; GPUS=8; BENCHMARK="in.snap.exaalt"; cd /benchmark;
/opt/lammps/tpl/openmpi/bin/mpirun -np $GPUS /opt/lammps/bin/lmp -k on g $GPUS -sf kk -pk
kokkos cuda/aware on neigh half comm device neigh/qeq full newton on -v x $SIZE -v y $SIZE
-v z $SIZE -v steps $STEPS -in $BENCHMARK -nocite
```

How to compute the Figure of Merit (FOM)

From the completed `Workloads` list, select the completed `LAMMPS` job. Select the `STDOUT` tab to view the `stdout` logs. The product of the `number of atoms` and `timesteps/s` rate is the FOM. Here's an example `stdout` log from LJ run. The computed FOM is **816,175,360 atoms*timesteps/s (87808000 atoms * 9.295 timesteps/s)**.

```
-> Loop time of 64.5532 on 8 procs for 600 steps with 87808000 atoms
-> Performance: 4015.291 tau/day, 9.295 timesteps/s
```

Guide for CP2K Application

Select CP2K Docker Application.

Navigate to the `Application Configuration` screen.

For MI250 Queue

Copy the following code block in the `Run Script` text box to run `H2O-DFT-LS-NREPS2` benchmark on an 8-GPU compute node.

```
CPU_COUNT=96; RANKS=12; GPU_TYPE=MI250;
benchmark H2O-DFT-LS-NREPS2 --arch VEGA90A --gpu-type $GPU_TYPE --rank-stride
8 --omp-thread-count 8 --ranks $RANKS --gpu-count 8 --cpu-count
$CPU_COUNT --output /home/aac/H2O-DFT-LS-NREP2-8GPU
```

Copy the following code block in the `Run Script` text box to run `32-H2O-RPA` benchmark on an 8-GPU compute node.

```
CPU_COUNT=96; RANKS=12; GPU_TYPE=MI250;
benchmark 32-H2O-RPA --arch VEGA90A --gpu-type $GPU_TYPE --rank-stride 8
--omp-thread-count 8 --ranks $RANKS --gpu-count 8 --cpu-count $CPU_COUNT
--output /home/aac/32-H2O-RPA-8GPU
```

For MI210 Queue

Copy the following code block in the `Run Script` text box to run `H2O-DFT-LS-NREPS2` benchmark on an 8-GPU compute node.

```
CPU_COUNT=128; RANKS=16; GPU_TYPE=MI210;
benchmark H2O-DFT-LS-NREPS2 --arch VEGA90A --gpu-type $GPU_TYPE --rank-stride
8 --omp-thread-count 8 --ranks $RANKS --gpu-count 8 --cpu-count
$CPU_COUNT --output /home/aac/H2O-DFT-LS-NREP2-8GPU
```

Copy the following code block in the `Run Script` text box to run `32-H2O-RPA` benchmark on an 8-GPU compute node.

```
CPU_COUNT=128; RANKS=16; GPU_TYPE=MI210;
benchmark 32-H2O-RPA --arch VEGA90A --gpu-type $GPU_TYPE --rank-stride 8
--omp-thread-count 8 --ranks $RANKS --gpu-count 8 --cpu-count $CPU_COUNT
--output /home/aac/32-H2O-RPA-8GPU
```

How to Extract the Figure of Merit (FOM)

The output files from the CP2K benchmark run can be found under `/home/aac`

To find the FOM for `H2O-DFT-LS-NREPS2` benchmark:

```
grep "CP2K   " /home/aac/H2O-DFT-LS-NREP2-*
```

Here's an example output that shows **10.940** as the FOM.

```
CP2K                                                1  1.0      0.012      0.029    10.940    10.940
```

To find the FOM for `32-H2O-RPA` sub-benchmark:

```
grep -E 'Input.*H2O.*inp|CP2K   ' /home/aac/32-H2O-RPA-*
```

Guide for RELION Application

Select the `RELION` Docker Application

Navigate to the `Application Configuration` Screen

For MI210 Queue

Copy the following code block in the `Run Script` text to run the `2d benchmark` on an 8-GPU compute node.

```
run-benchmark --class 2d -g 8 -n 33 -j 3 -p 10 --iters 25 -i /dataset
```

Copy the following code block in the `Run Script` text box to run the `3d benchmark` on an 8-GPU compute node.

```
run-benchmark --class 3d -g 8 -n 9 -j 10 -p 10 --iters 25 -i /dataset
```

For MI250 Queue

Copy the following code block in the `Run Script` text to run the `2d benchmark` on an 8-GPU compute node.

```
run-benchmark --class 2d -g 8 -n 33 -j 2 -p 30 --iters 25 -i /dataset
```

Copy the following code block in `the Run Script` text box to run the `3d benchmark` on an 8-GPU compute node.

```
run-benchmark --class 3d -g 8 -n 9 -j 10 -p 10 --iters 25 -i /dataset
```

How to Extract the Figure of Merit (FOM)

From the completed `Workloads` list, select the completed `RELION` job. Select the `STDOUT` tab to view the `stdout` logs. The Elapsed Time (FOM) gives the FOM. Here's an example output that shows the FOM as **1518 seconds**.

```
Elapsed Time (FOM): 1518 seconds
```