

HPC Sample Job: NAMD

[Expand all](#) [Collapse all](#)

Overview

NAMD can be run using four main classes of resources:

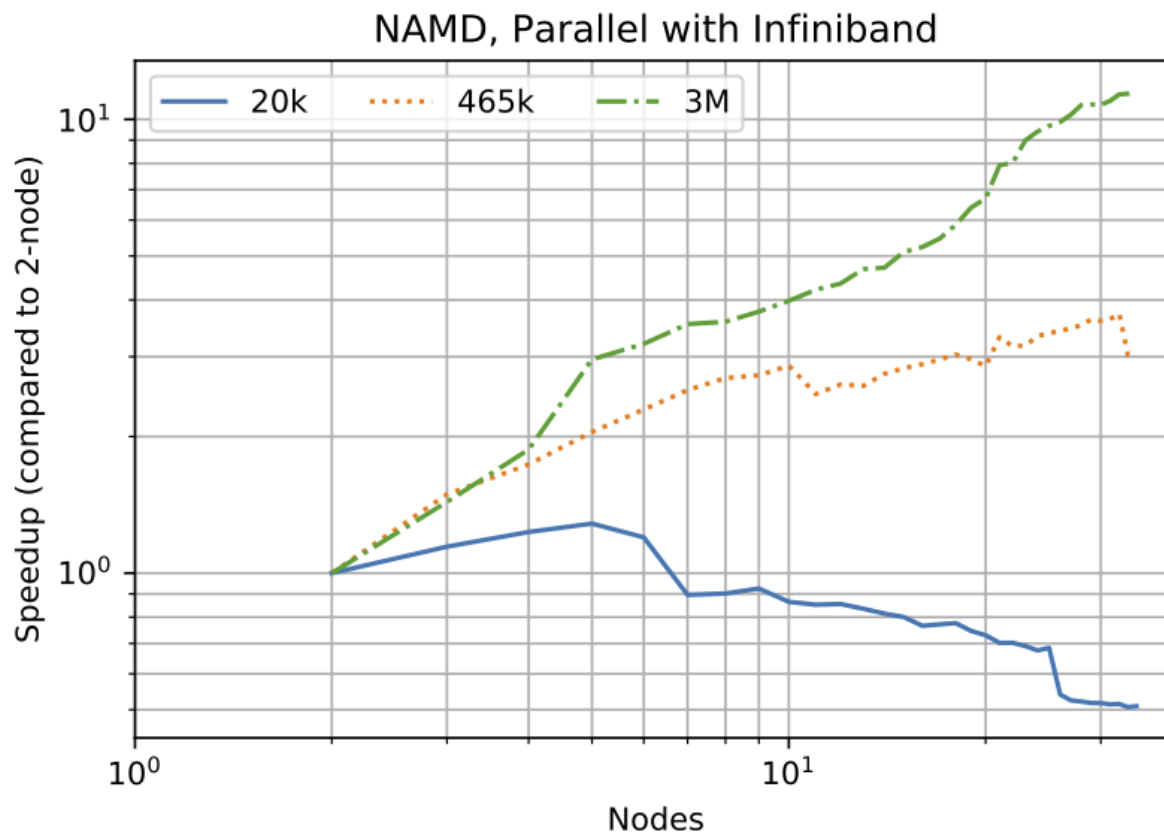
- small jobs using CPU cores in a single computer,
- small jobs using CPU cores and 1-2 GPUs in a single computer,
- large jobs using CPU cores in multiple computers connected over an Infiniband network, and
- large jobs using CPU cores and 1-2 GPUs per computer in multiple computers connected over an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

Though NAMD can use multiple computers connected over an Ethernet network, it is not recommended on the HPC, since the HPC processor performance is bottlenecked by the Ethernet bandwidth and latency.

To reduce the frequency of syntax errors or other problems for new NAMD users, we've written shell helper functions to help run NAMD in each of the four cases. In general, you'll only have to request resources in your Slurm script and run the helper functions to set up and run the job.

Input files for all the examples below are from the [HECBioSim](#) project's benchmark suite.

Additionally, the number of cores required for optimal performance is correlated to the size of the model. For example, the smallest HECBioSim model (20k atoms) reaches peak performance when using 5 28-core nodes in parallel, a medium model (465k atoms) reaches peak performance when using roughly 30 28-core nodes (with greatly diminishing returns after 10 28-core nodes), and the largest model (3M atoms) had consistent performance increases up to at least 34 28-core nodes.



Refer to the [Slurm Quick Start User Guide](#) for more information on Slurm scripts.

- [Single-computer, non-GPU NAMD Job](#)
- [Single-computer, GPU-enabled NAMD Job](#)
- [Multi-computer, Infiniband, non-GPU NAMD Job](#)
- [Multi-computer, Infiniband, GPU-enabled NAMD Job](#)
- [Appendix: contents of namd_functions](#)

Single-computer, non-GPU NAMD Job

Working from the HECBioSim 20k atom model, edit the file `bench.in` to ensure that the variable `new` is set to a value:

Section of NAMD input file

```
set new 20k_multicore
```

Create a Slurm job script named `namd_multicore.sh` to run the job:

namd_multicore.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=28

INPUT=bench.in
OUTPUT=bench.out
source /cm/shared/apps/namd/namd_functions
namd_setup # loads modules and sets up nodelist as needed
namd_run # runs charmrun or namd2 as needed using ${INPUT} and ${OUTPUT}
```

Submit the job from the login node with the command `sbatch namd_multicore.sh`, and when the job completes, you should have several new files named the same as your new variable setting (`20k_multicore`, in this case) and `bench.out`, containing the NAMD output.

Final directory listing for NAMD multicore job

```
[renfro@login NAMD]$ ls -lt
total 9720
-rw-r--r-- 1 renfro domain users 36460 Sep 1 20:23 bench.out
-rw-r--r-- 1 renfro domain users 470524 Sep 1 20:23 20k_multicore.vel
-rw-r--r-- 1 renfro domain users 2353676 Sep 1 20:23 20k_multicore.dcd
-rw-r--r-- 1 renfro domain users 470524 Sep 1 20:23 20k_multicore.coor
-rw-r--r-- 1 renfro domain users 1488 Sep 1 20:23 20k_multicore.xst
-rw-r--r-- 1 renfro domain users 261 Sep 1 20:23 20k_multicore.xsc
-rw-r--r-- 1 renfro domain users 470524 Sep 1 20:23 20k_multicore.vel.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 1 20:23 20k_multicore.coor.BAK
-rw-r--r-- 1 renfro domain users 262 Sep 1 20:23 20k_multicore.xsc.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 1 20:23 20k_multicore.vel.old
-rw-r--r-- 1 renfro domain users 470524 Sep 1 20:23 20k_multicore.coor.old
-rw-r--r-- 1 renfro domain users 262 Sep 1 20:23 20k_multicore.xsc.old
-rw-r--r-- 1 renfro domain users 1680 Sep 1 20:17 FFTW_NAMD_2.12_Linux-x86_64-multicore.txt
-rw-r--r-- 1 renfro domain users 0 Sep 1 20:17 slurm-1031.out
-rw-r----- 1 renfro domain users 1659 Sep 1 20:16 bench.in
-rw-r--r-- 1 renfro domain users 124 Sep 1 20:16 namd_multicore.sh
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.vel
-rw-r--r-- 1 renfro domain users 225 Dec 4 2013 relres.xsc
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.coor
-rw-r--r-- 1 renfro domain users 187048 Nov 14 2011 par_all27_prot_lipid.prm
-rw-r--r-- 1 renfro domain users 1548870 Nov 14 2011 solvated.pdb
-rw-r--r-- 1 renfro domain users 2013967 Nov 14 2011 solvated.psf
```

Single-computer, GPU-enabled NAMD Job

Working from the HECBioSim 20k atom model, edit the file `bench.in` to ensure that the variable `new` is set to a value:

Section of NAMD input file

```
set new 20k_gpu
```

Create a Slurm job script named `namd_gpu.sh` to run the job:

namd_gpu.sh

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=16
#SBATCH --partition=gpu
#SBATCH --gres=gpu:2

INPUT=bench.in
OUTPUT=bench.out
source /cm/shared/apps/namd/namd_functions
namd_setup # loads modules and sets up nodelists as needed
namd_run # runs charmrun or namd2 as needed using ${INPUT} and ${OUTPUT}
```

Submit the job from the login node with the command `sbatch namd_gpu.sh`, and when the job completes, you should have several new files named the same as your new variable setting (`20k_gpu`, in this case) and `bench.out`, containing the NAMD output.

Final directory listing for NAMD GPU job

```
[renfro@login NAMD]$ ls -lt
total 2104540
-rw-r--r-- 1 renfro domain users 38291 Sep 1 21:26 bench.out
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:26 20k_gpu.vel
-rw-r--r-- 1 renfro domain users 2353676 Sep 1 21:26 20k_gpu.dcd
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:26 20k_gpu.coor
-rw-r--r-- 1 renfro domain users 1479 Sep 1 21:26 20k_gpu.xst
-rw-r--r-- 1 renfro domain users 261 Sep 1 21:26 20k_gpu.xsc
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:26 20k_gpu.vel.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:26 20k_gpu.coor.BAK
-rw-r--r-- 1 renfro domain users 262 Sep 1 21:26 20k_gpu.xsc.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:26 20k_gpu.vel.old
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:26 20k_gpu.coor.old
-rw-r--r-- 1 renfro domain users 265 Sep 1 21:26 20k_gpu.xsc.old
-rw-r--r-- 1 renfro domain users 1681 Sep 1 21:24 FFTW_NAMD_2.12_Linux-x86_64-multicore-CUDA.txt
-rw-r--r-- 1 renfro domain users 0 Sep 1 21:23 slurm-1033.out
-rw-r----- 1 renfro domain users 1653 Sep 1 21:20 bench.in
-rw-r--r-- 1 renfro domain users 269 Sep 1 21:20 namd_gpu.sh
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.vel
-rw-r--r-- 1 renfro domain users 225 Dec 4 2013 relres.xsc
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.coor
-rw-r--r-- 1 renfro domain users 187048 Nov 14 2011 par_all127_prot_lipid.prm
-rw-r--r-- 1 renfro domain users 1548870 Nov 14 2011 solvated.pdb
-rw-r--r-- 1 renfro domain users 2013967 Nov 14 2011 solvated.psf
```

Multi-computer, Infiniband, non-GPU NAMD Job

Working from the HECBioSim 20k atom model, edit the file `bench.in` to ensure that the variable `new` is set to a value:

Section of NAMD input file

```
set new 20k_infiniband
```

Create a Slurm job script named `namd_infiniband.sh` to run the job:

namd_infiniband.sh

```
#!/bin/bash
#SBATCH --nodes=5
#SBATCH --ntasks-per-node=28

INPUT=bench.in
OUTPUT=bench.out
source /cm/shared/apps/namd/namd_functions
namd_setup # loads modules and sets up nodelists as needed
namd_run # runs charmrun or namd2 as needed using ${INPUT} and ${OUTPUT}
```

Submit the job from the login node with the command `sbatch namd_infiniband.sh`, and when the job completes, you should have several new files named the same as your new variable setting (`20k_infiniband`, in this case) and `bench.out`, containing the NAMD output.

Final directory listing for NAMD Infiniband job

```
[renfro@login1 NAMD]$ ls -lt
total 2104544
-rw-r--r-- 1 renfro domain users 38258 Sep 1 21:53 bench.out
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:53 20k_infiniband.vel
-rw-r--r-- 1 renfro domain users 2353676 Sep 1 21:53 20k_infiniband.dcd
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:53 20k_infiniband.coor
-rw-r--r-- 1 renfro domain users 1485 Sep 1 21:53 20k_infiniband.xst
-rw-r--r-- 1 renfro domain users 264 Sep 1 21:53 20k_infiniband.xsc
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:53 20k_infiniband.vel.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:53 20k_infiniband.coor.BAK
-rw-r--r-- 1 renfro domain users 265 Sep 1 21:53 20k_infiniband.xsc.BAK
-rw-r--r-- 1 renfro domain users 262 Sep 1 21:53 20k_infiniband.xsc.old
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:53 20k_infiniband.vel.old
-rw-r--r-- 1 renfro domain users 470524 Sep 1 21:53 20k_infiniband.coor.old
-rw-r--r-- 1 renfro domain users 1682 Sep 1 21:49 FFTW_NAMD_2.12_Linux-x86_64-ibverbs-smp.txt
-rw-r--r-- 1 renfro domain users 190 Sep 1 21:49 nodelist.1034
-rw-r--r-- 1 renfro domain users 0 Sep 1 21:49 slurm-1034.out
-rw-r--r-- 1 renfro domain users 429 Sep 1 21:49 namd_infiniband.sh
-rw-r----- 1 renfro domain users 1660 Sep 1 21:47 bench.in
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.vel
-rw-r--r-- 1 renfro domain users 225 Dec 4 2013 relres.xsc
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.coor
-rw-r--r-- 1 renfro domain users 187048 Nov 14 2011 par_all127_prot_lipid.prm
-rw-r--r-- 1 renfro domain users 1548870 Nov 14 2011 solvated.pdb
-rw-r--r-- 1 renfro domain users 2013967 Nov 14 2011 solvated.psf
```

Multi-computer, Infiniband, GPU-enabled NAMD Job

Before running a NAMD job requiring multiple nodes and GPU support, make sure to edit the `.bashrc` file in your home directory to include the line

Section of `.bashrc`

```
module load cuda80/toolkit
```

to ensure that the CUDA libraries and paths are available by default. This line is not required for other types of NAMD jobs, but is critical for multi-computer GPU-enabled NAMD jobs.

Working from the HECBioSim 20k atom model, edit the file `bench.in` to ensure that the variable `new` is set to a value:

Section of NAMD input file

```
set new 20k_infiniband_gpu
```

Create a Slurm job script named `namd_infiniband_gpu.sh` to run the job:

namd_infiniband_gpu.sh

```
#!/bin/bash
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=16
#SBATCH --partition=gpu
#SBATCH --gres=gpu:2
INPUT=bench.in
OUTPUT=bench.out
source /cm/shared/apps/namd/namd_functions
namd_setup # loads modules and sets up nodelists as needed
namd_run # runs charmrun or namd2 as needed using ${INPUT} and ${OUTPUT}
```

Submit the job from the login node with the command `sbatch namd_infiniband_gpu.sh`, and when the job completes, you should have several new files named the same as your new variable setting (`20k_infiniband_gpu`, in this case) and `bench.out`, containing the NAMD output.

Final directory listing for NAMD Infiniband GPU job

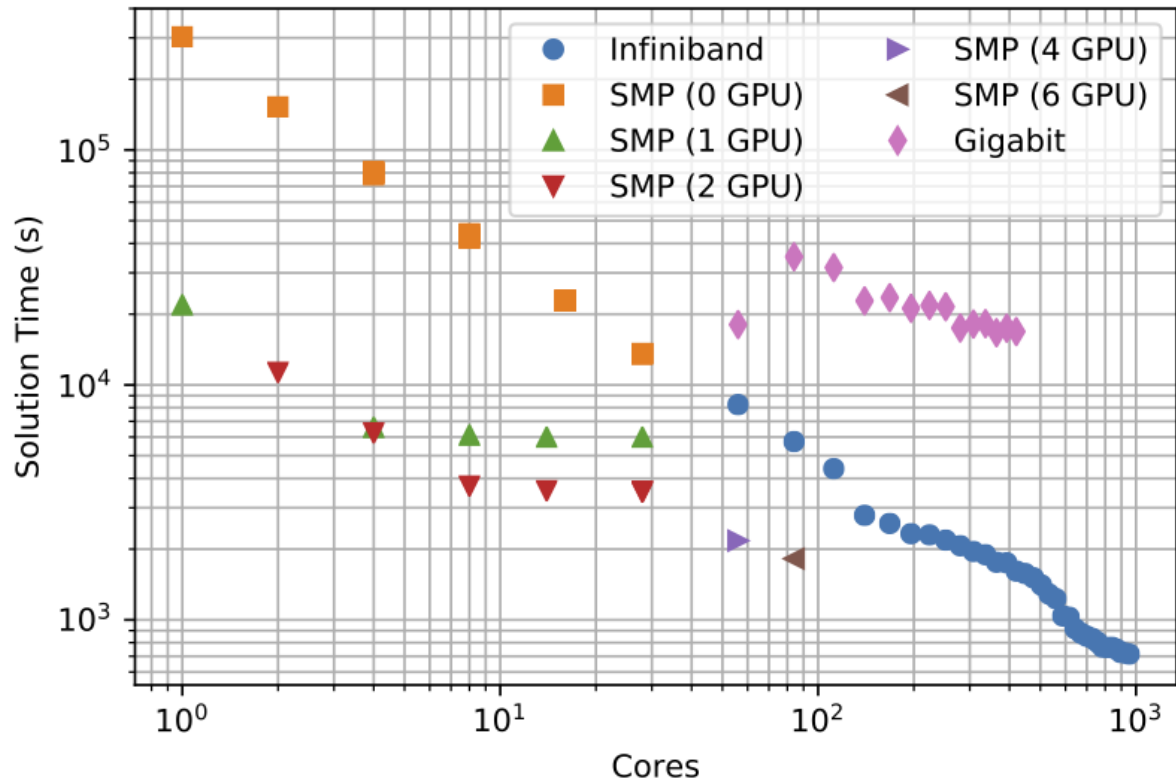
```
[renfro@login NAMD]$ ls -lt
total 9740
-rw-r--r-- 1 renfro domain users 41540 Sep 2 13:02 bench.out
-rw-r--r-- 1 renfro domain users 470524 Sep 2 13:02 20k_infiniband_gpu.vel
-rw-r--r-- 1 renfro domain users 2353676 Sep 2 13:02 20k_infiniband_gpu.dcd
-rw-r--r-- 1 renfro domain users 470524 Sep 2 13:02 20k_infiniband_gpu.coor
-rw-r--r-- 1 renfro domain users 1467 Sep 2 13:02 20k_infiniband_gpu.xst
-rw-r--r-- 1 renfro domain users 264 Sep 2 13:02 20k_infiniband_gpu.xsc
-rw-r--r-- 1 renfro domain users 265 Sep 2 13:02 20k_infiniband_gpu.xsc.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 2 13:02 20k_infiniband_gpu.vel.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 2 13:02 20k_infiniband_gpu.coor.BAK
-rw-r--r-- 1 renfro domain users 470524 Sep 2 13:01 20k_infiniband_gpu.vel.old
-rw-r--r-- 1 renfro domain users 470524 Sep 2 13:01 20k_infiniband_gpu.coor.old
-rw-r--r-- 1 renfro domain users 265 Sep 2 13:01 20k_infiniband_gpu.xsc.old
-rw-r--r-- 1 renfro domain users 1681 Sep 2 12:58 FFTW_NAMD_2.12_Linux-x86_64-ibverbs-smp-CUDA.txt
-rw-r--r-- 1 renfro domain users 82 Sep 2 12:58 nodelist.1046
-rw-r--r-- 1 renfro domain users 0 Sep 2 12:58 slurm-1046.out
-rw-r--r-- 1 renfro domain users 1664 Sep 2 12:56 bench.in
-rw-r--r-- 1 renfro domain users 555 Sep 2 12:38 namd_infiniband_gpu.sh
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.vel
-rw-r--r-- 1 renfro domain users 225 Dec 4 2013 relres.xsc
-rw-r--r-- 1 renfro domain users 470524 Dec 4 2013 relres.coor
-rw-r--r-- 1 renfro domain users 187048 Nov 14 2011 par_all27_prot_lipid.prm
-rw-r--r-- 1 renfro domain users 1548870 Nov 14 2011 solvated.pdb
-rw-r--r-- 1 renfro domain users 2013967 Nov 14 2011 solvated.psf
```

Performance Comparison for Large NAMD Job

As seen in the figure below, a large (3M atom) NAMD model shows consistent performance increases scaling up to at least 34 28-core nodes, as long as the nodes are connected with Infiniband networking. When connected with Ethernet networking (diamond markers on the figure below), 15 28-core nodes runs only as fast as 2 28-core nodes, and neither runs as fast as a single 28-core node. This behavior is worse on smaller models, as each core and node has fewer calculations to perform before needing to communicate for the next step of the solution.

Additionally, using 1-2 GPUs per GPU-enabled server results in runtimes equivalent to using 3-5 times as many non-GPU-enabled servers.

NAMD, 3M atom model



Appendix: contents of namd_functions

The following is the contents of the file `/cm/shared/apps/namd/namd_functions` as of 2017/09/04. Provided for reference, generally users won't need to do anything but use the functions as documented.

Contents of /cm/shared/apps/namd/namd_functions

```
function namd_make_nodelist() {
  > nodelist.${SLURM_JOBID}
  for n in `echo $SLURM_NODELIST | scontrol show hostnames`; do
    LINE="host ${n}.hpc.tntech.edu ++cpus ${SLURM_CPUS_ON_NODE}"
    echo "${LINE}" >> nodelist.${SLURM_JOBID}
  done
  CHARMRUN_ARGS="++p ${SLURM_NTASKS} ++ppn ${SLURM_CPUS_ON_NODE}"
  CHARMRUN_ARGS="${CHARMRUN_ARGS} ++nodelist nodelist.${SLURM_JOBID}"
}

function namd_setup() {
  CHARMRUN_ARGS=""
  NAMD_ARGS=""
  if [ "${GPU_DEVICE_ORDINAL}" == "NoDevFiles" ]; then
    # No GPUs reserved
    if [ "${SLURM_NNODES}" -gt 1 ]; then
      # multiple nodes without GPUs
      module load namd/ibverbs-smp
      namd_make_nodelist
      NAMD_ARGS=""
    else
      # single node without GPUs
      module load namd/multicore
      NAMD_ARGS="+p${SLURM_NTASKS}"
    fi
  else
    # GPUs reserved
    NAMD_ARGS="+devices ${GPU_DEVICE_ORDINAL}"
    if [ "${SLURM_NNODES}" -gt 1 ]; then
      # multiple nodes with GPUs
      module load namd/ibverbs-smp-cuda
      namd_make_nodelist
    else
      # single node with GPUs
      module load namd/cuda
      NAMD_ARGS="${NAMD_ARGS} +p${SLURM_NTASKS}"
    fi
  fi
}

function namd_run() {
  if [ "${SLURM_NNODES}" -gt 1 ]; then
    charmrun `which namd2` ${CHARMRUN_ARGS} ${NAMD_ARGS} \
      ${INPUT} >& ${OUTPUT}
  else
    namd2 ${NAMD_ARGS} ${INPUT} >& ${OUTPUT}
  fi
}
```

How helpful was this information?



✎

Your Rating:  Results:  70 rates