TCBG Internal GPU Tutorial - NAMD

Jim Phillips

March 4, 2013

Outline

- NAMD 2.9 (present)
  - What is accelerated?
  - What is disabled?
  - What is different?
  - Performance
  - Use in TCBG queues
  - Use on TCBG desktops
  - Use on Cray XK
- NAMD 2.10 (future)
  - Replica exchange
  - Kepler optimization
- Beyond NAMD 2.10

What is accelerated?

Accelerated

- Short-range non-bonded
  - Cutoff or with PME
  - w/ or w/o energy calculation
- Implicit solvent
- NVIDIA GPUs only

Not Accelerated

- Bonded terms
- PME reciprocal sum
- Integration
- Rigid bonds
- Grid forces
- Collective variables
- Etc.

What is disabled?

Disabled

- Alchemical (FEP and TI)
- Locally enhanced sampling
- Tabulated energies
- Drude (nonbonded Thole)
- Go forces
- Pairwise interaction
- Pressure profile

Not Disabled

- Memory optimized builds
- Conformational free energy
- Collective variables
- Grid forces
- Steering forces
- Almost everything else

What is different?

- Forces
  - Slightly less accurate than CPU
    - Different interpolation scheme, single precision
    - Also affects pressure calculation
- Energies
  - Don't match forces as closely as on CPU
  - Constant for interactions less than 1 Angstrom
    - Seems to be causing minimizer issues
    - Velocity-quenching minimizer should work better

Performance

What to expect

- 1 GPU = 12 CPU cores
  - Depending on CPU and GPU
- Scaling to 10K atoms/GPU
  - Assuming fast network
- Must use smp/multicore
  - Many cores share each GPU
  - Use multicore for single node
  - At most one process per GPU

Why it may be worse

- Weak GPU (e.g., laptop)
- Too few CPU cores used
- Coarse-grained simulation
- Too few atoms/GPU
- Limited by network
- Limited by MPI (use ibverbs)
- Limited by special features

Use in TCBG queues

- qsub –q gpu *script*
  - 24 hours, one node (-q gpu_short for 30 minutes)
  - 8 nodes available, 2 GPUs and 12 cores per node
  - Multi-node jobs not supported
- /usr/local/bin/namd2 *config* >& *output*
  - Recognizes queue environment, uses all cores/GPUs
  - Run older versions with, e.g., –version 2.8b3
- Check jobs with qstat
  - “qstat –u *” to show jobs of all users
  - “qstat –q gpu –u*” to show jobs running on GPU nodes

Use on TCBG desktops

- /usr/local/bin/namd2 –gpu +p*cores script*
- Automatically uses all (non-tiny) GPUs (max 1/core)
  - But runs CPU version if –gpu is not specified
  - Use "+devices *device,device,…*" for specific GPUs
  - Use nvidia_smi to see available GPUs
- Does not automatically use all cores
  - Same for the CPU version (i.e., without –gpu)
  - Use all cores or leave one free for interaction
  - May want to use hyperthreads as well
  - "cat /proc/cpuinfo" or "numactl –show" to see available

Use on Cray XK (Blue Waters/Titan)

- Must use SMP with one process per node
- Must use UGNI (gemini_gni) network layer
  - MPI SMP is much slower for NAMD
- Recommend only 8 threads per node
  - Bulldozer processor has 16 cores arranged in pairs
- See NAMD_scripts in my home directories
  - aprun –n *nodes* –r 1 –N 1 –d 9 */path/to/*namd2 +ppn 8 +pemap 0-14:2 +commap 7 *config* >& *output*

NAMD 2.10: Replica exchange

- NAMD 2.9 only supported on MPI builds
  - Unable to scale GPU replicas to multiple nodes
- Now available on MPI, UGNI, and BlueGene/Q
  - GPU replicas can scale to multiple Cray XK nodes
  - Critical for proposed simulations on Titan
  - Development funded by Blue Waters project
  - Still not supported on Infiniband (ibverbs)

NAMD 2.10: Kepler optimization

- Current NAMD kernel design is from 2007
- Kepler is current NVIDIA GPU architecture
  - Used on Titan and Blue Waters
  - Adds new capabilities relevant for MD codes
- Designing and optimizing new kernel for Kepler
  - New version should be faster and scale better
  - Force/energy interpolation closer to CPU version
    - Minimizer should work as well as on CPU
  - Backport to Fermi, possibly earlier if possible

Beyond NAMD 2.10

Sooner

- Platforms:
  - Intel Xeon Phi (MIC)
  - CPU vector instructions
- Performance:
  - PME reciprocal sum on GPU
  - Bonded forces on GPU
- Features:
  - **Based on TCBG project needs**
  - Alchemical free energy?

Later

- Platforms:
  - OpenCL (for AMD GPUs)
- Performance:
  - Grid forces
  - Integration
  - Rigid bonds
- Features:
  - Infrequently used
  - Inherent performance limits