Getting Started

Resources

- http://manual.gromacs.org/documentation/current/index.html
- http://manual.gromacs.org/documentation/5.1/user-guide/mdrun-performance.html

Building

So there seems to be a lot of build options that you can choose from, some of the notable ones are

- **-DCMAKE_C_COMPILER**=xxx equal to the name of the C99 Compiler you wish to use (or the environment variable CC)
- **-DCMAKE_CXX_COMPILER**=xxx equal to the name of the C++98 compiler you wish to use (or the environment variable CXX)
- **-DGMX_MPI**=on to build using MPI support (generally good to combine with building only mdrun)
- **-DGMX_GPU**=on to build using nvcc to run using NVIDIA CUDA GPU acceleration or an OpenCL GPU
- -DGMX_USE_OPENCL=on to build with OpenCL support enabled. GMX_GPU must also be set.
- -DGMX_SIMD=xxx to specify the level of SIMD support of the node on which GROMACS will run
- **-DGMX_BUILD_MDRUN_ONLY**=on for building only mdrun, e.g. for compute cluster back-end nodes
- -DGMX_DOUBLE=on to build GROMACS in double precision (slower, and not normally useful)
- -DCMAKE_PREFIX_PATH=xxx to add a non-standard location for CMake to search for libraries, headers or programs
- -DCMAKE_INSTALL_PREFIX=xxx to install GROMACS to a non-standard location (default /usr/local/gromacs)
- -DBUILD_SHARED_LIBS=off to turn off the building of shared libraries to help with static linking
- **-DGMX_FFT_LIBRARY**=xxx to select whether to use fftw3, **mkl** or fftpack libraries for FFT support
- -DCMAKE_BUILD_TYPE=Debug to build GROMACS in debug mode

▼ **Build on k40 GPU nodes → V1**

```
wget http://ftp.gromacs.org/pub/gromacs/gromacs-2020.1.tar.gz
tar -xzf gromacs-2020.1.tar.gz

cd gromacs-2020.1

# need intel 18, because gromacs doesn't support 19
module load cmake/3.14.3
module load gcc/5.5.0
source /home/mbeukman/intel/2018/parallel_studio_xe_2018.4.057/psxevars.sh
module load cuda/10.0

mkdir build
cd build

cmake .. -DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON \
-DCMAKE_INSTALL_PREFIX=/home/mbeukman/otherBenches/gromacs/v1GPU/install \
-DCMAKE_C_COMPILER=icc -DCMAKE_CXX_COMPILER=icpc -DGMX_MPI=on -DGMX_GPU=on -DGMX_FFT_LIBRARY=mkl 2>&1 | tee cmake.log
make -j28 2>&1 | tee make.log
```

```
make check 2>&1 | tee makecheck.log
make install 2>&1 | tee makeinstall.log
```

▼ Notes

```
# Important stuff

# "-DCMAKE_C_COMPILER=xxx" equal to the name of the C99 Compiler you
#  wish to use (or the environment variable "CC")

# "-DCMAKE_CXX_COMPILER=xxx" equal to the name of the C++98 compiler
#  you wish to use (or the environment variable "CXX")

# "-DGMX_MPI=on" to build using MPI support (generally good to
#  combine with building only mdrun)

# "-DGMX_GPU=on" to build using nvcc to run using NVIDIA CUDA GPU
#  acceleration or an OpenCL GPU

# "-DGMX_USE_OPENCL=on" to build with OpenCL support enabled.
#  "GMX_GPU" must also be set.

# "-DGMX_SIMD=xxx" to specify the level of SIMD support of the node
#  on which GROMACS will run

# "-DGMX_BUILD_MDRUN_ONLY=on" for building only mdrun, e.g. for
#  compute cluster back-end nodes

# "-DGMX_DOUBLE=on" to build GROMACS in double precision (slower,
#  and not normally useful)

# "-DCMAKE_PREFIX_PATH=xxx" to add a non-standard location for CMake
#  to search for libraries, headers or programs

# "-DCMAKE_INSTALL_PREFIX=xxx" to install GROMACS to a non-standard
#  location (default "/usr/local/gromacs")

# "-DBUILD_SHARED_LIBS=off" to turn off the building of shared
#  libraries to help with static linking

# "-DGMX_FFT_LIBRARY=xxx" to select whether to use "fftw3", "mkl" or
#  "fftpack" libraries for FFT support

# "-DCMAKE_BUILD_TYPE=Debug" to build GROMACS in debug mode

# source env
#module load intel/2019.5.281
```

▼ It might be better to only build mdrun, since the build took a few hours for me.

▼ **Build V2** → Thread MPI, only viable for a single node (might work with DGX)

```
wget http://ftp.gromacs.org/pub/gromacs/gromacs-2020.1.tar.gz
tar -xzf gromacs-2020.1.tar.gz

cd gromacs-2020.1

# need intel 18, because gromacs doesn't support 19
module load cmake/3.14.3
module load gcc/5.5.0
source /home/mbeukman/intel/2018/parallel_studio_xe_2018.4.057/psxevars.sh
module load cuda/10.0

mkdir build
cd build

cmake .. -DGMX_BUILD_OWN_FFTW=ON -DREGRESSIONTEST_DOWNLOAD=ON \
-DCMAKE_INSTALL_PREFIX=/home/mbeukman/otherBenches/gromacs/v2GPU/install \
-DCMAKE_C_COMPILER=icc -DCMAKE_CXX_COMPILER=icpc -DGMX_THREAD_MPI=on -DGMX_GPU=on -DGMX_FFT_LIBRARY=mkl 2>&1 | tee cmake.lo
make -j28 2>&1 | tee make.log
make check 2>&1 | tee makecheck.log
make install 2>&1 | tee makeinstall.log
```

Running

Consider the benchmark: http://www.gromacs.org/@api/deki/files/216/=ethanol_solvation.tgz

Running gromacs usually consists of two steps,

1. Preprocess data using the grommp tool.

   This follows the structure: `gmx_mpi grommp -f X.mdp -c X.gro -p X.top -o X.tpr`

   ```
   gmx_mpi grompp -f ethanol_direct.1.mdp -c ethanol.gro -p ethanol.top \
   -o ethanol.X.tpr -maxwarn 10
   ```

2. Run the simulation to using the above data
   1. `-gpu_id` are the ids of the gpus that are available (on K40s, 0 and 1)

```
mpirun -np 4 gmx_mpi  mdrun -gpu_id 01 -s ethanol.X.tpr
```

3. Output

   The output contains the time info

```
Core t (s)   Wall t (s)        (%)
 Time:      887.625       31.714     2798.9
               (ns/day)    (hour/ns)
Performance:      108.981        0.220
```

Results

▼ First I source this env script

```
module load cmake/3.14.3
module load gcc/5.5.0
source /home/mbeukman/intel/2018/parallel_studio_xe_2018.4.057/psxevars.sh
module load cuda/10.0
export PATH=$PATH:/home/mbeukman/otherBenches/gromacs/v1GPU/install/bin
```

All these results were using the ethanol benchmark, with the nsteps set to 20000, on the k40s

Results

Aa Name	Description	# Nodes	# MPI Ranks	# OMP Threads Per Rank	# NS/day	# Core Time	# Wall Time
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -np 4 gmx_mpi mdrun -dlb auto -dd 0 -gpu_id 01 -s ethanol.X.tpr	1	4	7	102.596	943.23	33.687
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -np 4 gmx_mpi mdrun -gpu_id 01 -s ethanol.X.tpr	1	4	7	109.114	886.832	31.675
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -np 6 gmx_mpi mdrun -gpu_id 01 -s ethanol.X.tpr	1	6	4	239.822	345.868	14.411

Aa Name	Description	# Nodes	# MPI Ranks	# OMP Threads Per Rank	# NS/day	# Core Time	# Wall Time
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -np 8 gmx_mpi mdrun -gpu_id 01 -s ethanol.X.tpr	1	8	3	181.36	457.362	19.057
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -machinefile gmx_mpi mdrun -gpu_id 01 -s ethanol.X.tpr	2	8	7	243.545	794.686	14.191
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -machinefile gmx_mpi mdrun -gpu_id 01 -dlb yes -dd 0 -s ethanol.X.tpr -npme 6	2	18	3	273.556	682.236	12.634
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -machinefile gmx_mpi mdrun -gpu_id 01 -dlb auto -dd 0 -s ethanol.X.tpr -npme 6	2	14	4	337.285	573.816	10.247
/home/mbeukman/otherBenches/gromacs/v1GPU → K40 GPU above build script	mpirun -machinefile machinefile gmx_mpi mdrun -s ethanol.X.tpr -npme 6	2	14	4	343.028	564.211	10.075
/home/mbeukman/otherBenches/gromacs/v2GPU → K40 GPU Build V2	gmx mdrun -s ethanol.X.tpr -ntmpi 4 -gpu_id 01 -npme 2	1	4	7	401.525	241.007	8.608
V1	mpirun -machinefile machinefile gmx_mpi mdrun -s ethanol.X.tpr -npme 4	2	8	7	439.538	440.323	7.863
V1	mpirun -np 4 gmx_mpi mdrun -s ethanol.X.tpr -gpu_id 01 -npme 2	1	4	7	376.893	256.757	9.17
V1	mpirun -np 8 gmx_mpi mdrun -s ethanol.X.tpr -gpu_id 01 -npme 4	1	8	3	299.228	277.202	11.55
Untitled							

▼ The machinefile for multi node runs looks something like this:

```
gpu03:4
gpu04:4
```

Observations

- The GPU usage is never more than ~40%, and I'm not sure why that is.
  - **With 6 mpi ranks, the gpu usage was ~80% and doubled the performance**
- The CPU is mostly green
- Running on multiple nodes does not seem to net much performance gain, and I'm note sure why...