NHR FAU

Friedrich-Alexander-Universität Erlangen-Nürnberg

Gromacs — Best Practices

HPC Café, 8 March 2022

HPC Services, NHR@FAU

Gromacs — Timeline

- Version 1.0 published 1995[1]
- Functionality based on GROMOS[2]
- Programs are written in ANSI C, inner loops in Fortran, no MPI support yet
- Developed for a custom-designed 32-processor ring “GROMACS” (GROningen MAchine for Chemical Simulation)
- Available force fields: GROMOS and OPLS
- Peptide of 20 residues in water runs 100 time steps (0.2 ps) in one minute on a 32-i860 processor machine →12 ps per hour or 1 ns in 3 days.

1. H.J.C. Berendsen, D. van der Spoel, R. van Drunen. GROMACS: A message-passing parallel molecular dynamics implementation. 1995. Comput Phys Commun, 91:43–56.
2. W.E van Gunsteren and H.J.C. Berendsen, GROMOS (GROningen MOlecular Simulation package), BIOMOS B.V., Nijenborgh 4, 9747 AG Groningen, The Netherlands.

Gromacs — Timeline

- Version 3.0 published 2001[3] under GNU General Public License with MPI support and AMBER force field
- SIMD instructions on x86 processors
- Original PME implementation (same as in AMBER), but single precision
- Most time consuming part of the calculation: pairwise distances from particle coordinates (squaring the inter-particle vector produces $\mathbf{r}^2$)
- Construction of a special software routine for the reciprocal square root operation $x^{-1/2}$ to calculate $\mathbf{r}^{-1}$ directly from $\mathbf{r}^2$ (Coulomb) or the reciprocal operation 1/x when starting from $\mathbf{r}^2$ (Lennard-Jones)
- Multimedia instructions on processors use x^{-1} and $x^{-1/2}$ for lighting processing in games → table lookups implemented in hardware

3. E. Lindahl, B. Hess, D. van der Spoel. GROMACS 3.0: a package for molecular simulation and trajectory analysis. 2001. J Mol Model, 7:306–317.

Gromacs — Timeline

- Version 3.3[4] (2007) added ordered communication scheme (OpenMP-based multithreading) for scalability across multiple CPU nodes
- Version 4[5] published 2008: eighth-shell domain decomposition method + dynamic load-balancing with minimum communication + parallel version of LINCS + dedicated PME ranks
- Version 4.5[6] (2013): GPU port + ‘thread_MPI’ that implements MPI communication calls using multithreading for domain decomposition

4. C. Kutzner, D. van der Spoel, M. Fechner, E. Lindahl, U. W. Schmitt, B. L. de Groot, H. Grubmüller. Speeding Up Parallel GROMACS on High-Latency Networks. 2007. J Comput Chem, 28:2075–2084.
5. B. Hess, C. Kutzner, D. van der Spoel, E. Lindahl. GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation. 2008. J Chem Theory Comput, 4:435–447.
6. S. Pronk, S. Páll, R. Schulz , et al. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. 2013. Bioinformatics, 2(7):845–854.

Gromacs — Timeline

- Version 4.6[7] (2013): exclusion bitmask of pair-interactions between clusters of particles of fixed size → 15% performance increase
- Version 5.0[8,9] (2015): transition to C++98
- Version 2016[10]: removal of the twin-range multiple-time-stepping scheme (different time step for long-range nonbonded interactions)

7. S. Páll, B. Hess. A flexible algorithm for calculating pair interactions on SIMD architectures. 2013. Comput Phys Commun, 184(12):2641–2650.
8. S. Páll, M. J. Abraham, C. Kutzner, B.Hess, E. Lindahl. 2015. Tackling Exascale Software Challenges in Molecular Dynamics Simulations with GROMACS. Published in: Solving Software Challenges for Exascale, Print ISBN: 978-3-319-15975-1.
9. M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, E.Lindahl. 2015. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX, 1–2:19–25.
10. B. Hess, D. van der Spoel, M. J. Abraham, E. Lindahl. On The Importance of Accurate Algorithms for Reliable Molecular Dynamics Simulations. 2019. ChemRxiv. This content is a preprint and has not been peer-reviewed.

Gromacs — Timeline

- Version 2018: added GPU support for PME
- Version 2019: GPU offload for bonded interactions
- Version 2020[11]: GPU offload for updates and constraints
- Version 2021: Added a two-level multiple time-stepping scheme + SIMD acceleration for bonds in systems with H-bonds only constrained or no constraints + Allow offloading GPU update and constraints without direct GPU communication + Improved performance of the short-ranged non-bonded kernels by up to 12% on NVIDIA Ampere GPUs
- Version 2022: Improvements in GPU direct communication

11. S. Páll, A.Zhmurov, P. Bauer, M. Abraham, M. Lundborg, A. Gray, B. Hess, E Lindahl. 2020. Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS. J Chem Phys, 153:134110.

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to RTX2080Ti

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to RTX2080Ti

	1 GPU	2 GPUs	4 GPUs
Gromacs2019	134 ns/day	176 ns/day	230 ns/day
Gromacs2020	229 ns/day	188 ns/day	273 ns/day
Gromacs2021.1	233 ns/day	195 ns/day	294 ns/day
Gromacs2021.5	237 ns/day	212 ns/day	227 ns/day
Gromacs2022	234 ns/day	176 ns/day	199 ns/day

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculati… …080Ti

`-update` only works on multiple GPUs with update-groups

	1		4 GPUs
Gromacs2019	13…	…s/day	230 ns/day
Gromacs2020	2…9 ns/day	188 …s/day	273 ns/day
Gromacs2021.1	233 ns/day	195 ns/day	294 ns/day
Gromacs2021.5	237 ns/day	212 ns/day	227 ns/day
Gromacs2022	234 ns/day	176 ns/day	199 ns/day

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculati… …080Ti

`-update` only works on multiple GPUs with update-groups

	1	…s	4 GPUs
Gromacs2019	13…	…ns/day	230 ns/day
Gromacs2020	2… 9 ns/day	188 …s/day	273 ns/day
Gromacs2021.1	233 ns/day	195 ns/day	294 ns/day
Gromacs2021.5	237 ns/day	212 ns/day	227 ns/day
Gromacs2022	234 ns/day	176 ns/day	199 ns/day

Always use the fastest Gromacs version possible!

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to GPUs, Gromacs2021.1/.4
- `-update` does not work on multiple GPUs

	1 GPU	2 GPUs	4 GPUs	8 GPUs
A100	332 ns/day	247 ns/day	334 ns/day	n.a.
A40	320 ns/day	323 ns/day	383 ns/day	235 ns/day
RTX3080	266 ns/day	203 ns/day	281 ns/day	322 ns/day
RTX2080Ti	233 ns/day	195 ns/day	294 ns/day	n.a.

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to GPUs, Gromacs2021.1/.4
- `-update` does not work PUs

+ 2 ns
S(N)=0.25

	1 PU	2 GPUs	GPUs	8 GPUs
A100	332 ns/day	247 ns/day	334 ns/day	n.a.
A40	320 ns/day	323 ns/day	383 ns/day	235 ns/day
RTX3080	266 ns/day	203 ns/day	281 ns/day	322 ns/day
RTX2080Ti	233 ns/day	195 ns/day	294 ns/day	n.a.

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to GPUs, Gromacs2021.1/.4
- `-update` does not work[illegible]PUs

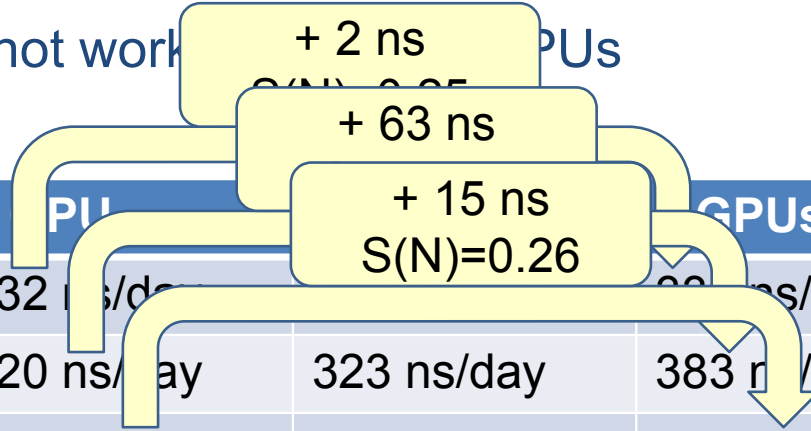
+ 2 ns

+ 63 ns
S(N)=0.3

	1 [illegible]PU		[illegible]GPUs	8 GPUs
A100	332 [illegible]s/day	247 ns/day	33[illegible] ns/day	n.a.
A40	320 ns/day	323 ns/day	383 ns/day	235 ns/day
RTX3080	266 ns/day	203 ns/day	281 ns/day	322 ns/day
RTX2080Ti	233 ns/day	195 ns/day	294 ns/day	n.a.

Gromacs — Case 1

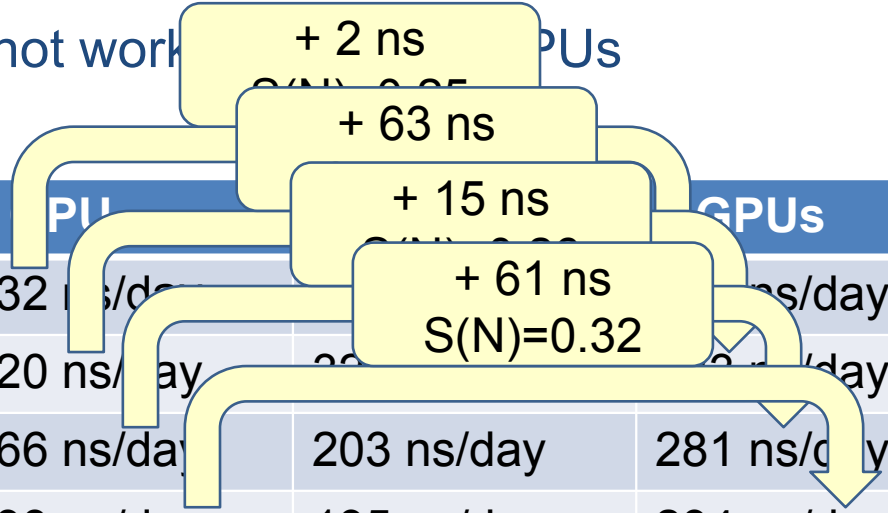
- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to GPUs, Gromacs2021.1/.4
- `-update` does not work[illegible]PUs



	1 [illegible]PU		[illegible]GPUs	8 GPUs
A100	332 [illegible]s/day		[illegible]ns/day	n.a.
A40	320 ns/[illegible]ay	323 ns/day	383 [illegible]/day	235 ns/day
RTX3080	266 ns/day	203 ns/day	281 ns/day	322 ns/day
RTX2080Ti	233 ns/day	195 ns/day	294 ns/day	n.a.

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to GPUs, Gromacs2021.1/.4
- `-update` does not work[illegible]PUs



	1 [illegible]PU	[illegible] GPUs	8 GPUs
A100	332 [illegible]s/day	[illegible]ns/day	n.a.
A40	320 ns/[illegible]ay	[illegible]day	235 ns/day
RTX3080	266 ns/day	203 ns/day	281 ns/[illegible]y	322 ns/day
RTX2080Ti	233 ns/day	195 ns/day	294 ns/day	n.a.

Gromacs — Case 1

- Standard all-atom MD simulation: protein in membrane with explicit water (65,209 atoms)
- All possible calculations offloaded to GPUs, Gromacs2021.1/.4

Don’t do this!

It’s a waste of resources.

Gromacs — Case 2

- Standard all-atom MD: protein inside a membrane surrounded by explicit water (80,289 atoms)
- Gromacs2021.1(RTX*)/.2(V100)/.3(A40)/.4(A100, Fritz), Gromacs2020.1 (HAWK, Meggie), all possible offloading to GPU, tuning needed on CPU

Gromacs — Case 2

- Standard all-atom MD: protein inside a membrane surrounded by explicit water (80,289 atoms)
- Gromacs2021.1(RTX*)/.2(V100)/.3(A40)/.4(A100, Fritz), Gromacs2020.1 (HAWK, Meggie), all possible offloading to GPU, tuning needed on CPU

	RTX2080Ti	V100	RTX3080	A40	A100
[ns/day]	169.3	153.1	191.2	194.8	247.1

Gromacs — Case 2

- Standard all-atom MD: protein inside a membrane surrounded by explicit water (80,289 atoms)
- Gromacs2021.1(RTX*)/.2(V100)/.3(A40)/.4(A100, Fritz), Gromacs2020.1 (HAWK, Meggie), all possible offloading to GPU, tuning needed on CPU

	RTX2080Ti	V100	RTX3080	A40	A100
[ns/day]	169.3	153.1	191.2	194.8	247.1

[ns/day]	1 node	2 nodes	4 nodes	6 nodes	12 nodes	16 nodes
Meggie	17.9	35.0	65.4	91,0	165.4	207.9
Fritz	117.6	217.5	370.9	510.0	n.a.	n.a.
HAWK	115.2	208.0	369.7	477.8	n.a.	n.a.

Gromacs — Case 2

- Standard all-atom MD: protein inside a membrane surrounded by explicit water (80,289 atoms)
- Gromacs2021.1(RTX*)/.2(V100)/.3(A40)/.4(A100, Fritz), Gromacs2020.1 (HAWK, Meggie), all possible offloading to GPU, tuning needed on CPU

	RTX2080Ti	V100	RTX3080	A40	A100
[ns/day]	169.3	153.1	191.2	194.8	247.1

[ns/day]	1 node	2 nodes	4 nodes	6 nodes	12 nodes	16 nodes
Meggie	17.9	35.0	65.4	91,0	165.4	207.9
Fritz	117.6	217.5	370.9	510.0	n.a.	n.a.
HAWK	115.2	208.0	369.7	477.8	n.a.	n.a.

Gromacs — Case 2

- Standard all-atom MD: protein inside a membrane surrounded by explicit water (80,289 atoms)
- Gromacs2021.1(RTX*)/.2(V100)/.3(A40)/.4(A100, Fritz), Gromacs2020.1 (HAWK, Meggie), all possible offloading to GPU, tuning needed on CPU

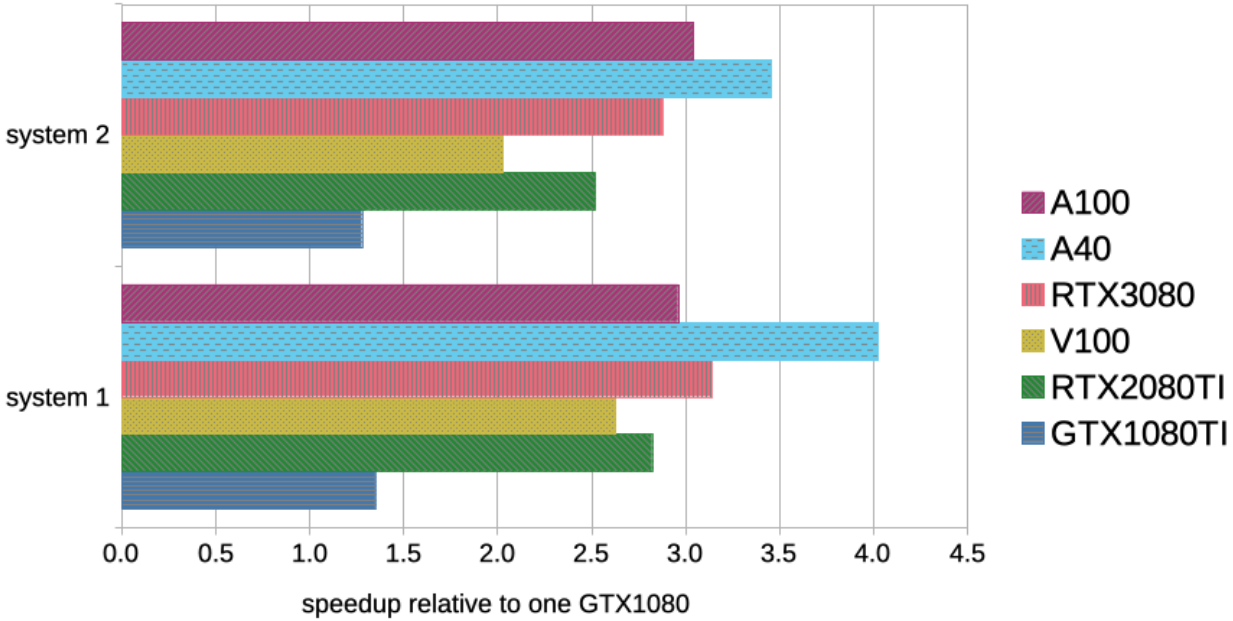
	RTX2080Ti	V100	RTX3080	A40	A100
[ns/day]	169.3	153.1	191.2	194.8	247.1

[ns/day]	1 node	2 nodes	4 nodes	8 nodes	12 nodes	16 nodes
Meggie	17.9	35.0	65.4	91,0	165.4	207.9
Fritz	117.6	217.5	370.9	510.0	n.a.	n.a.
HAWK	115.2	208.0	369.7	477.8	n.a.	n.a.

Gromacs — Small Systems

- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

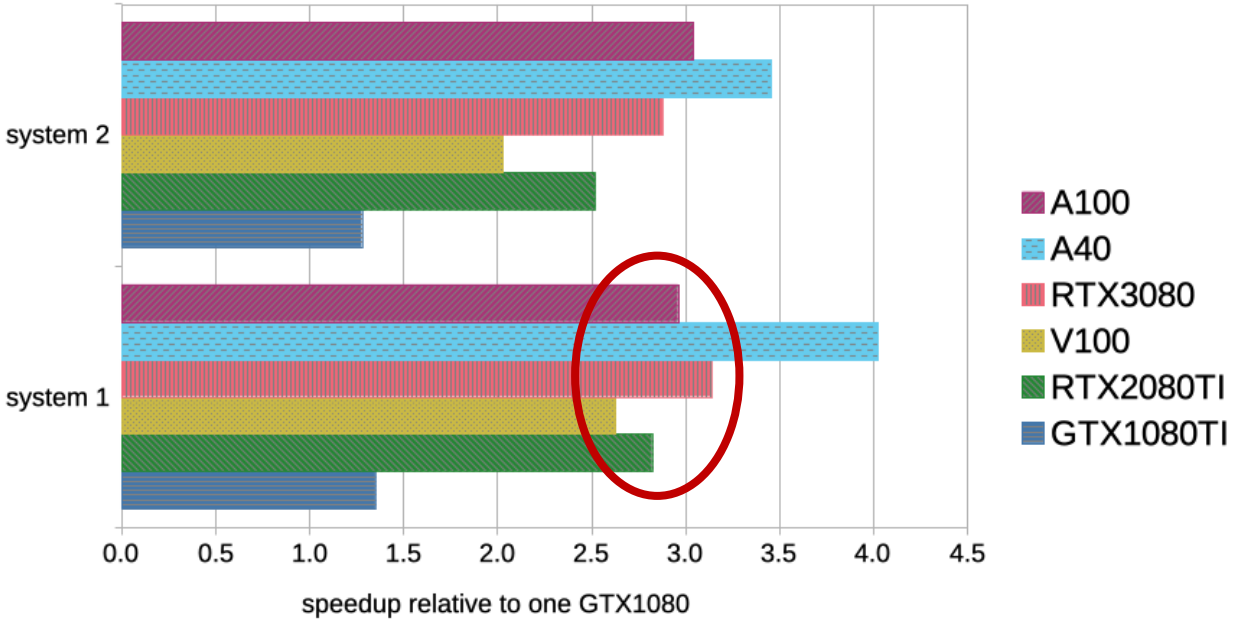
[ns/day]	system 1	system 2
GTX1080Ti	84.3	257.8
RTX2080Ti	176.2	508.4
V100	163.8	408.7
RTX3080	195.9	580.4
A40	251.0	696.6
A100	184.8	613.2



Gromacs — Small Systems

- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

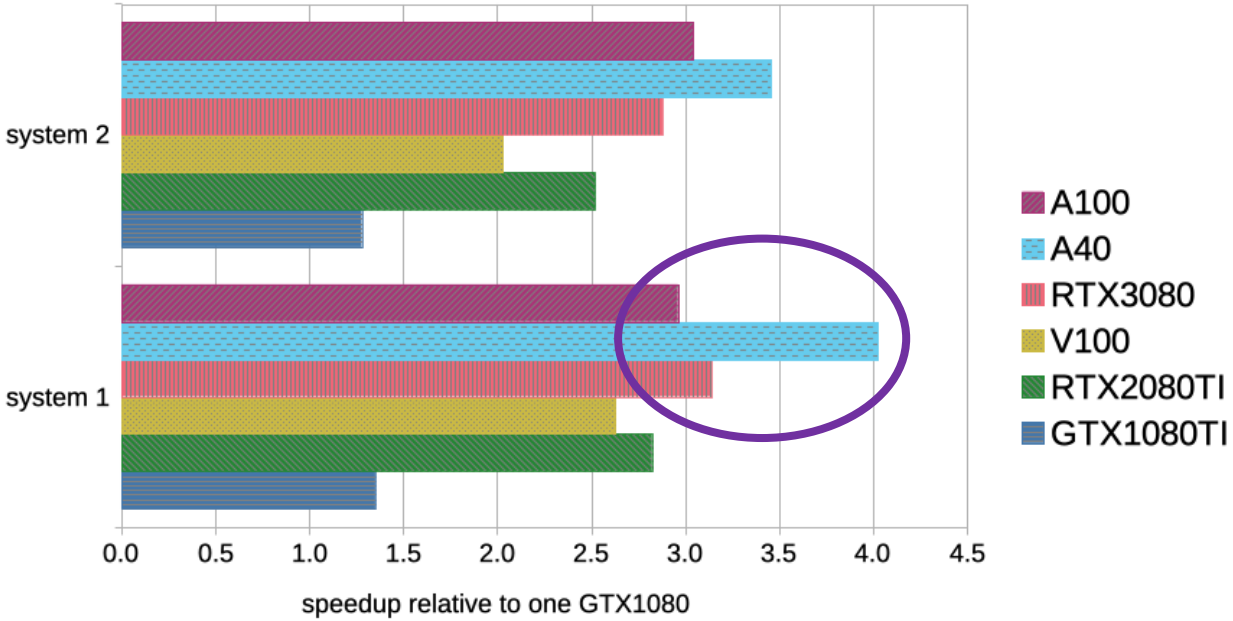
[ns/day]	system 1	system 2
GTX1080Ti	84.3	257.8
RTX2080Ti	176.2	508.4
V100	163.8	408.7
RTX3080	195.9	580.4
A40	251.0	696.6
A100	184.8	613.2



Gromacs — Small Systems

- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

[ns/day]	system 1	system 2
GTX1080Ti	84.3	257.8
RTX2080Ti	176.2	508.4
V100	163.8	408.7
RTX3080	195.9	580.4
A40	251.0	696.6
A100	184.8	613.2



Gromacs — Small Systems

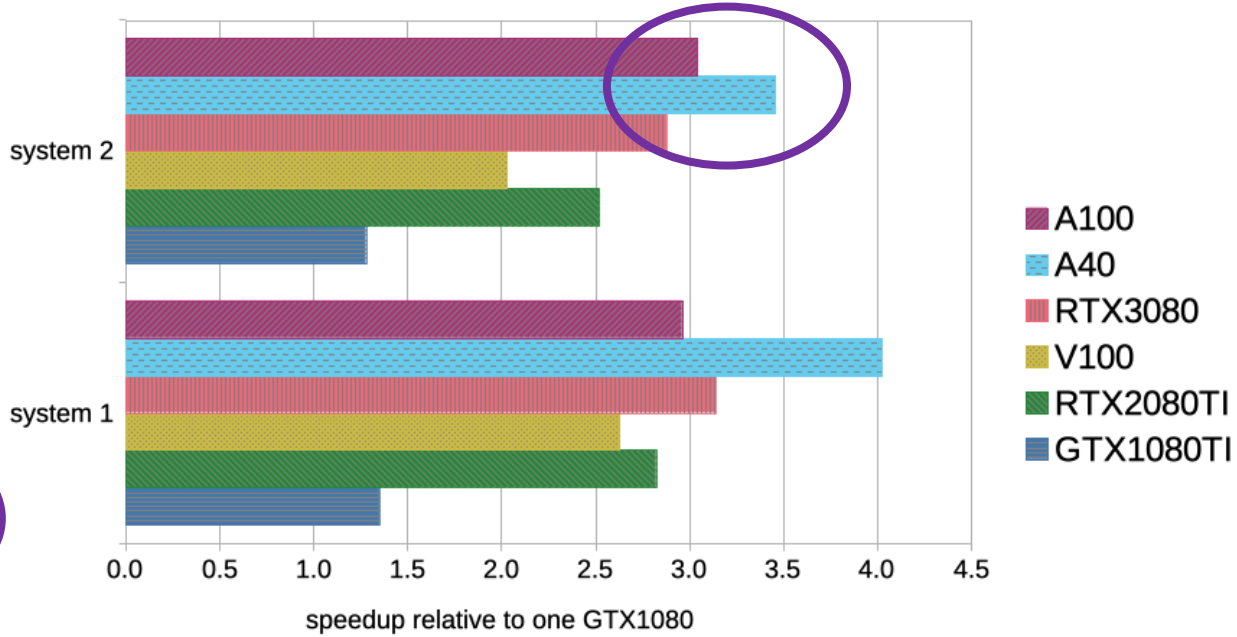
- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

[ns/day]	system 1	system 2
GTX1080Ti	84.3	257.8
RTX2080Ti	176.2	508.4
V100	163.8	408.7
RTX3080	195.9	580.4
A40	251.0	696.6
A100	184.8	613.2

Gromacs — Small Systems

- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

[ns/day]	system 1	system 2
GTX1080Ti	84.3	257.8
RTX2080Ti	176.2	508.4
V100	163.8	408.7
RTX3080	195.9	580.4
A40	251.0	696.6
A100	184.8	613.2

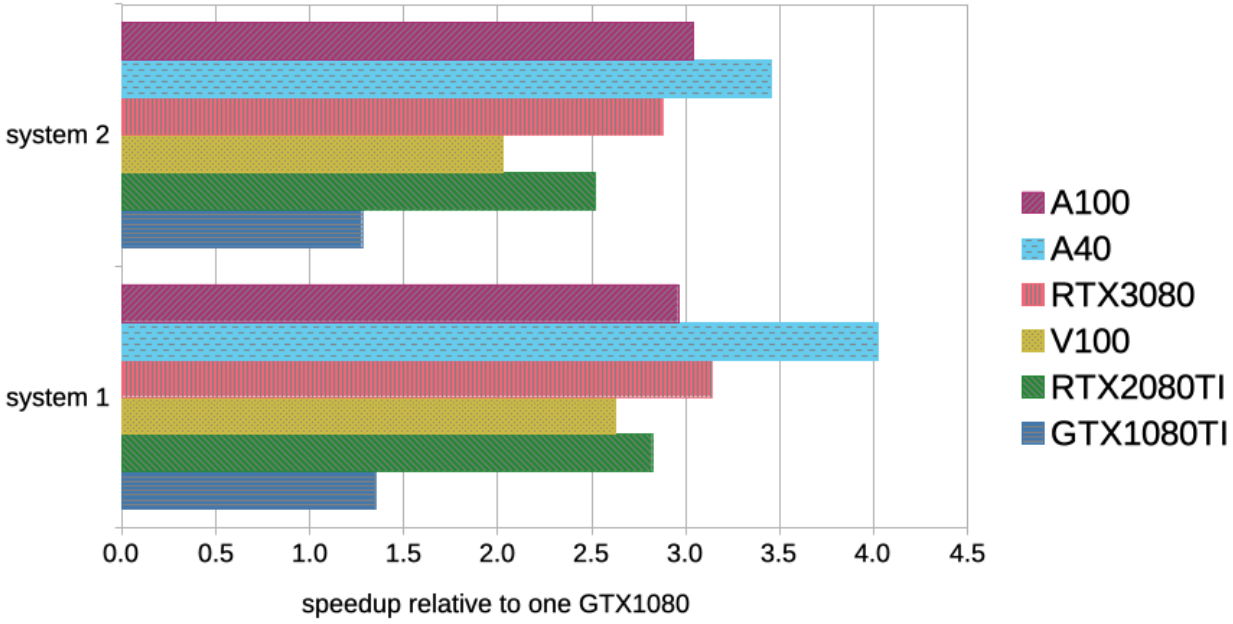


Gromacs — Small Systems

- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

[ns/day]	system 1	system 2
GTX1080Ti	84.3	257.8
RTX2080Ti	176.2	508.4
V100	163.8	408.7
RTX3080	195.9	580.4
A40	1.0	6
A100		2

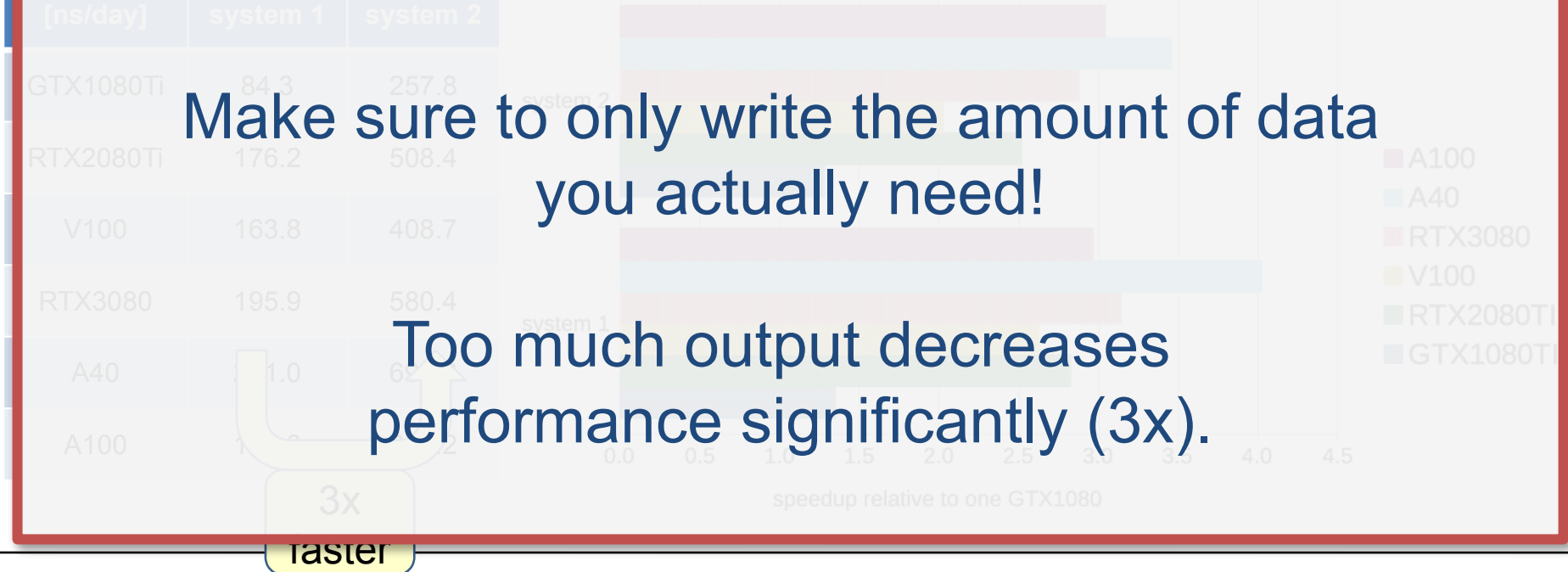
3x faster



Gromacs — Small Systems

- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

[ns/day]	system 1	system 2
GTX1080Ti	84.3	257.8
RTX2080Ti	176.2	508.4
V100	163.8	408.7
RTX3080	195.9	580.4
A40	1.0	62
A100	1	2



Make sure to only write the amount of data you actually need!

Too much output decreases performance significantly (3x).

Gromacs — Small Systems

- System 1: R-143a in hexane (20,248 atoms) with very high output rate
- System 2: a short RNA piece with explicit water (31,889 atoms)

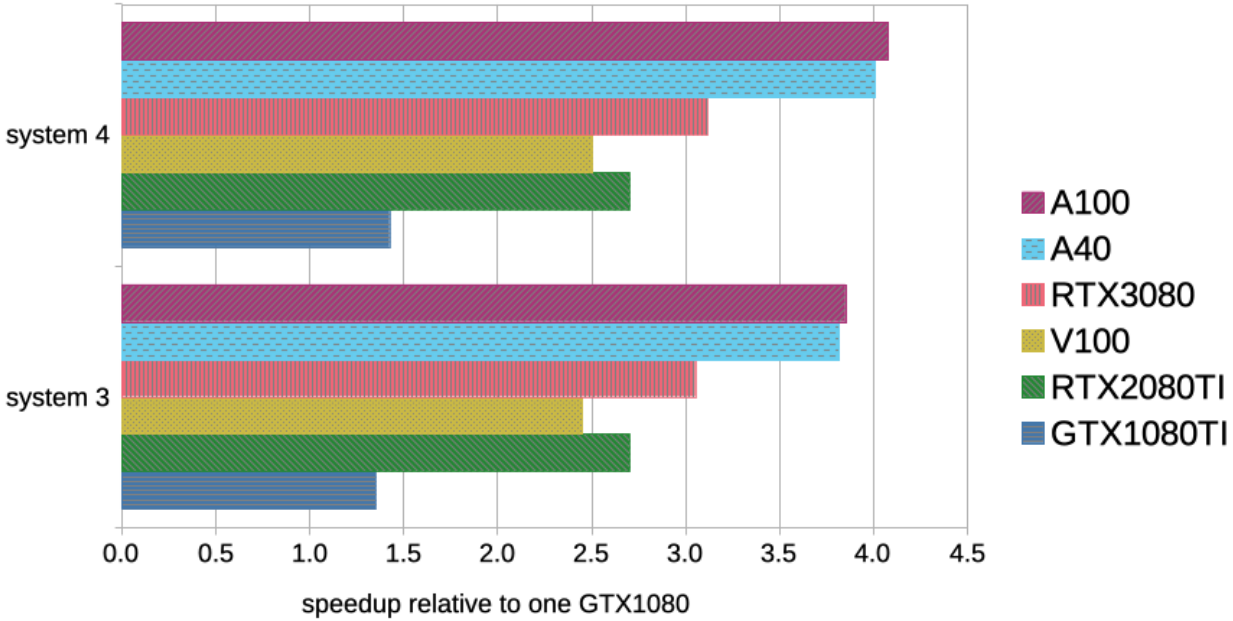
- Preferably use TinyGPU.

- Add `#SBATCH --gres=gpu:1` to your Slurm script to get a random GPU on TinyGPU without waiting in the queue for too long.

Gromacs — Medium Systems

- System 3: membrane protein surrounded by explicit water (80,289 atoms)
- System 4: a protein in explicit water (170,320 atoms)

[ns/day]	system 3	system 4
GTX1080Ti	84.6	41.8
RTX2080Ti	169.3	79.2
V100	153.1	73.3
RTX3080	191.2	91.3
A40	238.8	117.4
A100	241.2	119.5



Gromacs — Medium Systems

- System 3: membrane protein surrounded by explicit water (80,289 atoms)
- System 4: a protein in explicit water (170,320 atoms)

[ns/day]	system 3	system 4
GTX1080Ti	84.6	41.8
RTX2080Ti	169.3	79.2
V100	153.1	73.3
RTX3080	191.2	91.3
A40	238.8	117.4
A100	241.2	119.5

Gromacs — Medium Systems

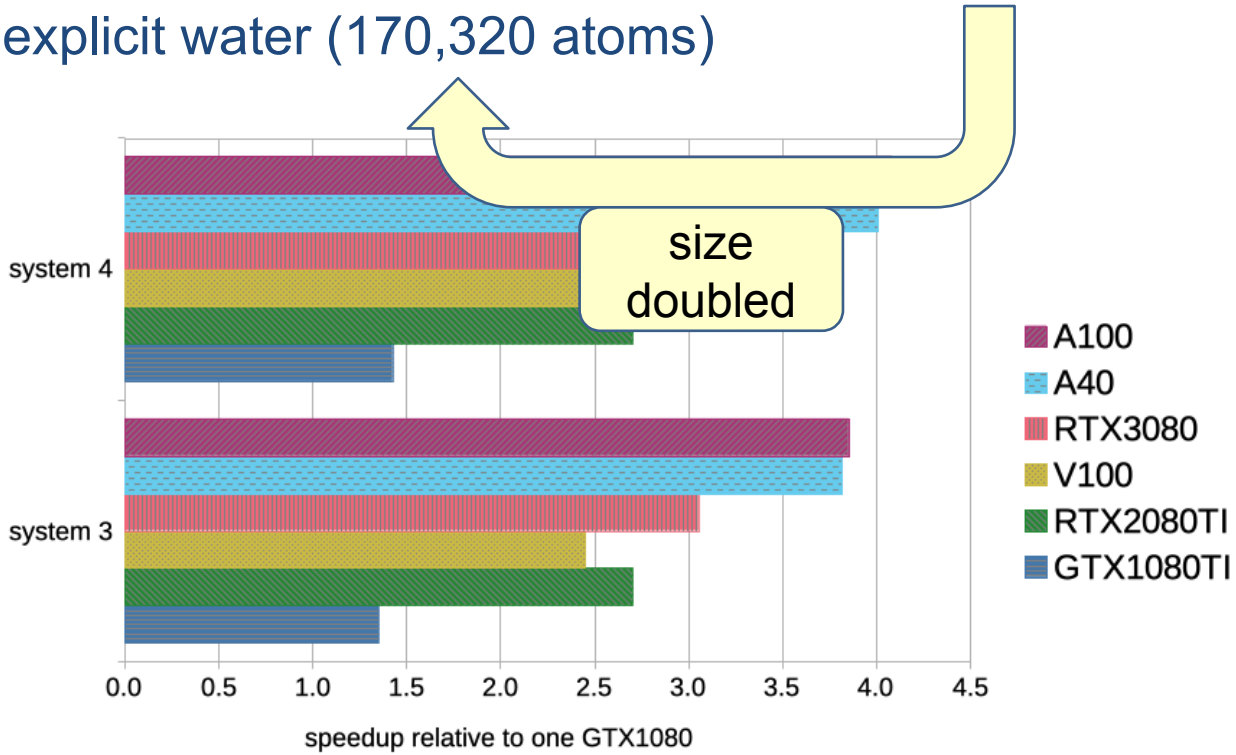
- System 3: membrane protein surrounded by explicit water (80,289 atoms)
- System 4: a protein in explicit water (170,320 atoms)

[ns/day]	system 3	system 4
GTX1080Ti	84.6	41.8
RTX2080Ti	169.3	79.2
V100	153.1	73.3
RTX3080	191.2	91.3
A40	238.8	117.4
A100	241.2	119.5

Gromacs — Medium Systems

- System 3: membrane protein surrounded by explicit water (80,289 atoms)
- System 4: a protein in explicit water (170,320 atoms)

[ns/day]	system 3	system 4
GTX1080Ti	84.6	41.8
RTX2080Ti	169.3	79.2
V100	153.1	73.3
RTX3080	191.2	91.3
A40	238.8	117.4
A100	241.2	119.5

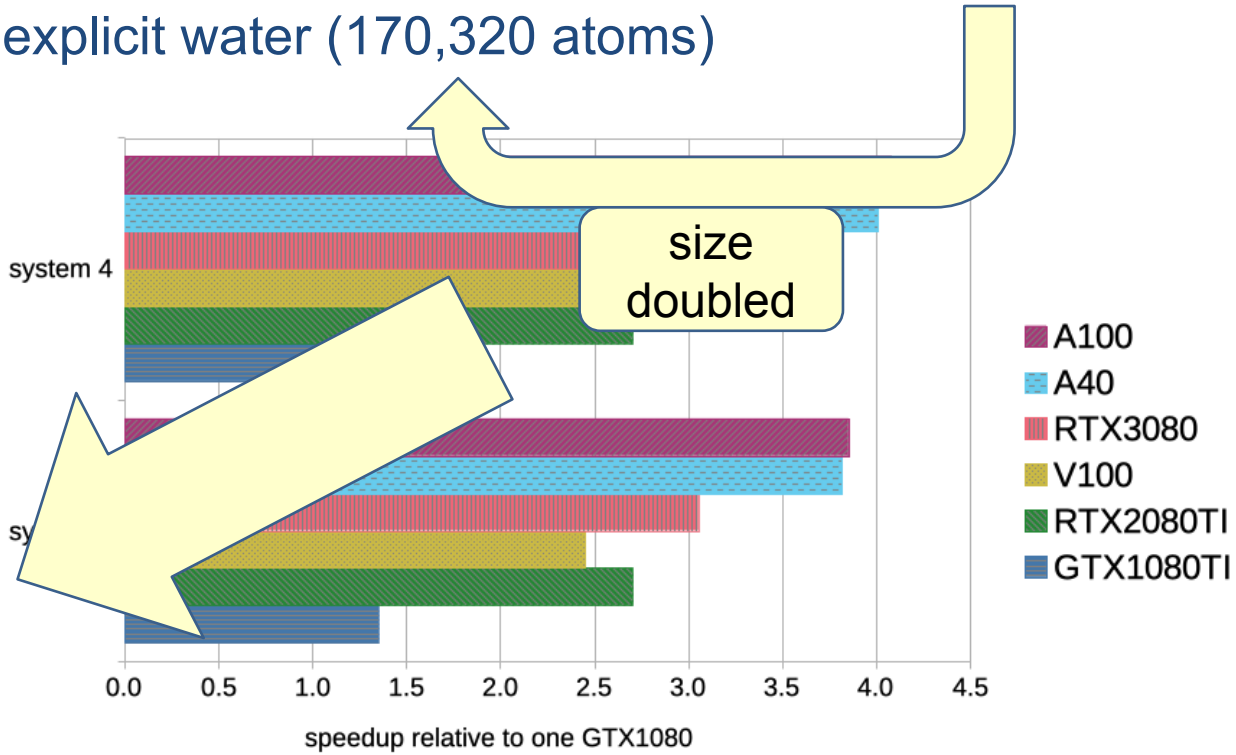


Gromacs — Medium Systems

- System 3: membrane protein surrounded by explicit water (80,289 atoms)
- System 4: a protein in explicit water (170,320 atoms)

[ns/day]	system 3	system 4
GTX1080Ti	84.6	41.8
RTX2080Ti	169.3	79.2
V100	153.1	73.3
RTX3080	191.2	91.3
A40	38.8	1 .4
A100	2	9.5

50% slower



Gromacs — Medium Systems

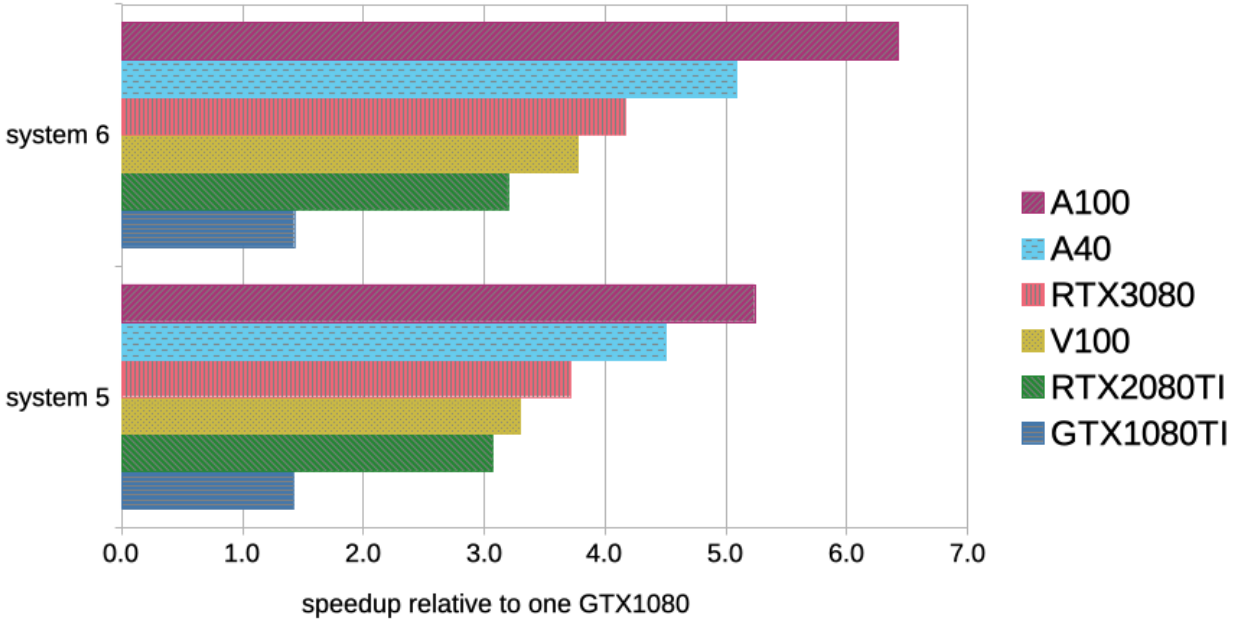
- System 3: membrane protein surrounded by explicit water (80,289 atoms)
- System 4: a protein in explicit water (170,320 atoms)

- Use TinyGPU or the A40s in Alex.
- Be aware that the larger number of CPU cores per GPU on the A40 and A100 nodes in Alex can make a decisive difference.
- Add `#SBATCH --gres=gpu:1` to your Slurm script to get a random GPU on TinyGPU without waiting in the queue for too long.

Gromacs — Large Systems

- System 5: protein membrane channel with explicit water (615,924 atoms)
- System 6: huge virus protein (1,066,628 atoms)

[ns/day]	system 5	system 6
GTX1080Ti	10.1	5.0
RTX2080Ti	21.8	11.2
V100	23.4	13.2
RTX3080	26.4	14.6
A40	32.0	17.8
A100	37.2	22.5



Gromacs — Large Systems

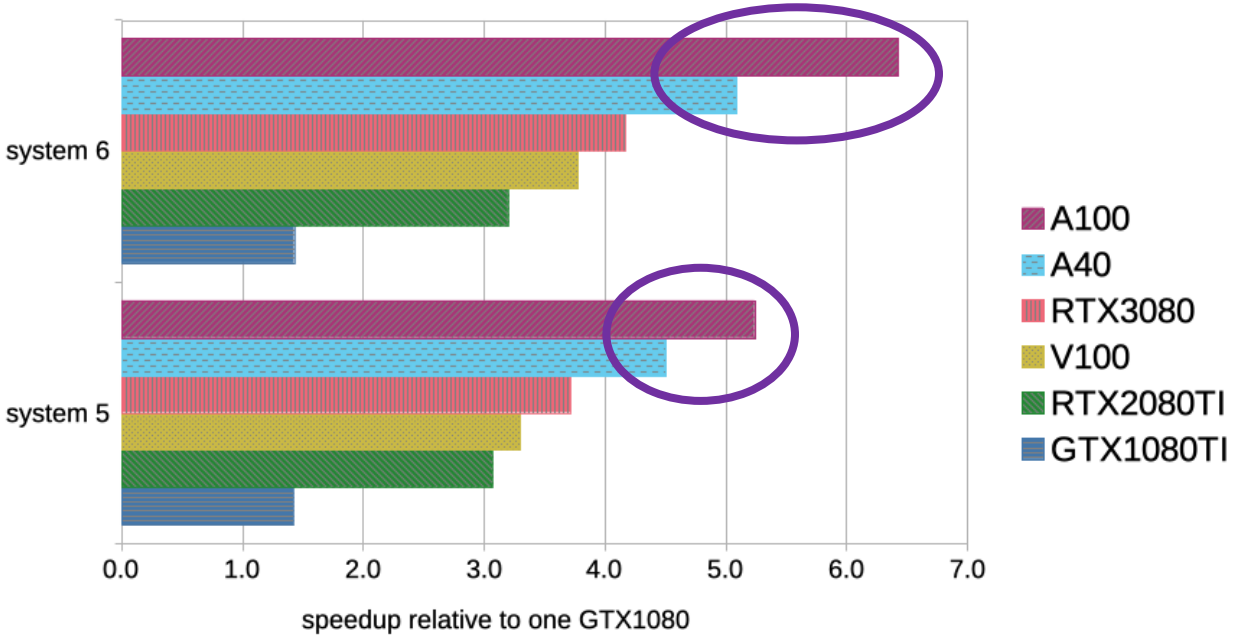
- System 5: protein membrane channel with explicit water (615,924 atoms)
- System 6: huge virus protein (1,066,628 atoms)

[ns/day]	system 5	system 6
GTX1080Ti	10.1	5.0
RTX2080Ti	21.8	11.2
V100	23.4	13.2
RTX3080	26.4	14.6
A40	32.0	17.8
A100	37.2	22.5

Gromacs — Large Systems

- System 5: protein membrane channel with explicit water (615,924 atoms)
- System 6: huge virus protein (1,066,628 atoms)

[ns/day]	system 5	system 6
GTX1080Ti	10.1	5.0
RTX2080Ti	21.8	11.2
V100	23.4	13.2
RTX3080	26.4	14.6
A40	32.0	17.8
A100	37.2	22.5



Gromacs — Large Systems

- System 5: protein membrane channel with explicit water (615,924 atoms)
- System 6: huge virus protein (1,066,628 atoms)

Perform benchmarks to find out which hardware performs best for you.

Check if performance increases with multiple GPUs.

Gromacs — Usage

- Find the newest Gromacs version by typing "`module avail gromacs`" on the target node.
- Load the module with "`module load gromacs/…`".

- Modules containing "`cuda`" can only be loaded on the GPU nodes!

- The MD engine is "`gmx mdrun`" for the thread-MPI version and
  "`srun gmx_mpi mdrun`" for the real MPI version.

- Add the run input file `my-simulation.tpr` with the "`-s`" flag.
- Give the output a nice name with "`-deffnm nice-simulation-name`".
- Don't use verbose output "`-v`" (only for debugging crashed simulations).

Gromacs — Usage on GPUs

```
gmx mdrun -s my-simulation.tpr -deffmn nice-simulation-name \
          -ntmpi $ntmpi -ntomp $ntomp -pin on -pinstride 1 \
          -nb gpu -pme gpu -bonded gpu -update gpu
```

- `ntmpi`: Number of thread-MPI threads = number of GPUs
- `ntomp`: Number of OpenMP threads per MPI rank = number of cores

- Slurm script for TinyGPU: `#SBATCH --gres=gpu:1`
- On Alex (A40, A100): `-ntomp 16`
- On TinyGPU (RTX2080Ti, RTX3080, V100): `-ntomp 8`
  (GTX1080, GTX1080Ti): `-ntomp 4`

Gromacs — Usage on multiple GPUs

Only for very large systems and when performance increase is real.

Add these variables for halo exchange and
optimized communication between the GPUs.

```
export GMX_GPU_DD_COMMS=true
export GMX_GPU_PME_PP_COMMS=true
(export GMX_GPU_FORCE_UPDATE_DEFAULT_GPU=true)
```

Gromacs — Usage on multiple GPUs

Only for very large systems and when performance increase is real.

Add these variables for halo exchange and optimized communication between the GPUs.

Please contact us for benchmarking.

```
export GMX_GPU_DD_COMMS=true
export GMX_GPU_PME_PP_COMMS=true
(export GMX_GPU_FORCE_UPDATE_DEFAULT_GPU=true)
```

Gromacs — Usage on CPUs with MPI support

- `gmx_mpi mdrun -s my-simulation.tpr -maxh $time -dlb yes \`
  `-npme $npme -ntomp $ntomp`

- **maxh**: Terminate after 0.99 times this time (hours) = set to walltime – 0.5

- **npme**: Number of separate ranks to be used for PME (usage of tune_pme can be tricky → get in touch with us)
- **ntomp**: Number of OpenMP threads per MPI rank to start (must be tested)

First try something like and then optimize parameters!

```
srun gmx_mpi mdrun -nsteps 200000 -s my-simulation.tpr
```

Gromacs — .mdp-Options

- Equilibrations cannot run on GPUs!

- Set the time step for integration dt [ps] to 2 fs: `dt = 0.002`
- Output control:
  - `nstxout = 250000 ; save coordinates every 500 ps`
  - `nstvout = 250000 ; save velocities every 500 ps`
  - `nstenergy = 250000 ; save energies every 500 ps`
  - `nstxtcout = 50000 ; xtc compr. trajectory output every 100 ps`
  - `nstlog = 500 ; update log file every 1 ps`

- Offload of “updates and constraints” to multiple GPUs only works with `constraints = h-bonds ; h bonds (+heavy h) constrained`
- Use `constraint_algorithm = lincs` (it’s faster).

Gromacs — Available GPUs

- In TinyGPU (Documentation):
  - NVIDIA GTX 1080 + 4 cores (Intel Xeon E5-2620v4 Broadwell, 2.1 GHz)
  - NVIDIA GTX 1080 Ti + 4 cores (Intel Xeon E5-2620v4 Broadwell, 2.1 GHz)
  - NVIDIA RTX 2080 Ti + (4 cores +SMT) (Intel Xeon Gold 6134 Skylake, 3.2 GHz)
  - NVIDIA Tesla V100 + (4 cores +SMT) (Intel Xeon Gold 6134 Skylake, 3.2 GHz)
  - NVIDIA Geforce RTX3080 + (4 cores +SMT) (Intel Xeon Gold 6226R, 2.9 GHz)
  - NVIDIA A100 + 32 cores (AMD Rome 7662, 2.0 GHz)

- In Alex (Documentation):
  - Nvidia A100 + 16 cores (AMD EPYC 7713 Milan, 2.0 GHz)
  - Nvidia A40 + 16 cores (AMD EPYC 7713 Milan, 2.0 GHz)

Gromacs — Where to store simulation data

- Data storage options:
  - HPC storage documentation
  - General work directory `$WORK` is either `$WOODYHOME` under the path `/home/woody/GROUPNAME/USERNAME` or `$SATURNHOME` under the path `/home/{saturn,titan}/GROUPNAME/USERNAME`
  - Vault accessible via `$HPCVAULT`
    available under the path `/home/vault/GROUPNAME/USERNAME`

- During the simulation:
  - Use either RAM disk or local HDD / SDD; both are accessible via `$TMPDIR`.
  - Always copy or move the generated data to your working directory.
  - Data in `$TMPDIR` is deleted after the job finishes!

Gromacs — Need help?

- Gromacs documentation:

  https://manual.gromacs.org/documentation/

- Gromacs special tips and tricks in our documentation:

https://hpc.fau.de/systems-services/systems-documentation-instructions/special-applications-and-tips-tricks/gromacs/

- MD virtual consultation hour:

  contact us hpc-support@fau.de