DESRES/TR--2021-01

Desmond/GPU Performance as of April 2021

Michael Bergdorf,[1] Avi Robinson-Mosher,[1] Xinyi Guo,[1] Ka-Hei Law,[1] and David E. Shaw[1,2,*]

Desmond/GPU Software Version 1.9.1 (Development Branch), March 22, 2021

Summary

Desmond/GPU is a code, written in CUDA C++, that is designed for the execution of molecular dynamics (MD) simulations of biochemical systems on NVIDIA graphics processing units (GPUs). This paper reports the performance achieved by Desmond/GPU, as of April 2021, running six benchmark systems on a range of GPU models and configurations.

Benchmark chemical systems and simulation parameters

Performance results were measured for six chemical systems that are commonly used for MD code benchmarking: DHFR [1], ApoA1 [2], F1-ATPase [2], a satellite tobacco mosaic virus (STMV) [2], a ribosome [3,4], and a 5 × 2 × 1 tiling of the STMV system (which we will refer to as 10× STMV). The characteristics of the benchmark systems are summarized in Table 1. The simulation parameters were chosen to optimize Desmond/GPU performance without compromising accuracy [5].

[1] D. E. Shaw Research, New York, NY 10036, USA.
[2] Department of Biochemistry and Molecular Biophysics, Columbia University, New York, NY 10032, USA.
[*] To whom correspondence should be addressed: David.Shaw@DEShawResearch.com.

All simulations were run with the *u*-series algorithm [5] for decomposing the electrostatic energy into a near part and a far part, and midtown splines [6] was used for the charge spreading and force interpolation steps of the far calculation.

The simulations were run with a non-bonded interaction cutoff of 9 Å in the NVT ensemble at a constant temperature of 298.15 K. We used a baseline time step of 2.5 fs, but we were also able to measure performance at a longer time step by using hydrogen mass repartitioning (HMR) [7]. HMR involves increasing the mass of all hydrogen atoms from 1 u to 4 u, while keeping the total mass of water molecules unchanged; this means that the oxygen atoms in water molecules are assigned a reduced mass of 10 u in an HMR system. This repartitioning reduces the frequency of the fastest degrees of freedom and allowed us to integrate the system with a time step of 4.0 fs.

Table 1. System characteristics and production parameters (the Far ES interval is the time-step interval at which the far electrostatic forces are evaluated).

System	# of atoms	System size (Å)	Far ES Grid size	Time step (fs)	Far ES interval
DHFR	23,558	62 × 62 × 62	32 × 32 × 32	2.5	3
				4.0	2
ApoA1	92,224	109 × 109 × 78	54 × 54 × 40	2.5	3
				4.0	2
F1-ATPase	327,506	179 × 132 × 133	88 × 64 × 66	2.5	3
				4.0	2
STMV	1,066,628	219 × 219 × 219	108 × 108 × 108	2.5	3
				4.0	2
Ribosome	2,180,503	280 × 280 × 280	140 × 140 × 140	2.5	3
				4.0	2
10× STMV	10,666,280	1088 × 435 × 218	528 × 216 × 108	2.5	3
				4.0	2

Hardware and operating environment

We ran the benchmarks on GPUs representing four different architectures—Pascal (GP102), Volta (GV100), Turing (TU102/TU104), and Ampere (GA100/GA102)—and using a variety of host systems. Details are listed in Table 2.

Table 2. Host systems used for benchmarking.

GPU	GPU architecture	Host system
GeForce GTX 1080 Ti	GP102	Supermicro - SYS-1028GQ-TRT
GeForce RTX 2080	TU104	Supermicro - SYS-4029GP-TRT
GeForce RTX 2080 Ti	TU102	Supermicro - SYS-4027GR-TR
GeForce RTX 3080	GA102	Dell Inc. - Precision 5820 Tower
GeForce RTX 3090	GA102	Supermicro - SYS-4029GP-TRT
V100 PCIe (16GB)	GV100	Supermicro - SYS-4027GR-TR
Quadro RTX 8000	TU102	Supermicro - SYS-4029GP-TRT2
Quadro RTX A6000	GA102	Supermicro - SYS-4029GP-TRT
A100 SXM4 (40GB)	GA100	Supermicro - AS -2124GQ-NART

We used various CPU architectures, including AMD EPYC, and Intel processors ranging from Broadwell to Cascade Lake. (The CPU used for any given simulation would not be expected to impact performance, as Desmond/GPU only uses a single CPU core to dispatch work to the GPUs.) All host systems ran under CentOS 7.7. Desmond/GPU was compiled using CUDA 11.1.1 and GCC 7.3.0. All simulations used NVIDIA driver version 460.32.03.

Results

In this section, Desmond/GPU simulation rates are reported in simulated nanoseconds per day of elapsed wall-clock time. Note that the version of Desmond/GPU used for running the benchmarks reported here does not support the use of more than one GPU in a single MD simulation, though replica exchange simulations can use more than one GPU. For a comparison of Desmond performance on CPUs versus GPUs, and for performance of previous versions of Desmond/GPU, we refer the reader to the preceding performance reports [9,10].

Table 3 reports the performance of Desmond/GPU for the systems listed in Table 1 using different GPUs. Table 4 reports the performance of Desmond/GPU when HMR is used to allow a time step of 4.0 fs. In both tables, darker shading of the cells indicates faster performance and the bolded values represent the fastest performance for each system.

Table 3. Benchmark results for simulations run with a 2.5-fs time step. Simulation rates are reported in simulated nanoseconds per day. Missing entries indicate that a benchmark did not run because it exceeded the memory capacity of the GPU.

GPU	Time step (fs)	DHFR	ApoA1	F1-ATPase	STMV	Ribosome	10× STMV
GeForce GTX 1080 Ti	2.5	871.4	236.2	50.8	16.0	8.6	
GeForce RTX 2080	2.5	1,082.3	310.7	73.3	21.6		
GeForce RTX 2080 Ti	2.5	1,307.5	416.9	103.9	29.1		
GeForce RTX 3080	2.5	1,524.3	525.5	143.3	42.1		
GeForce RTX 3090	2.5	1,697.2	**574.6**	166.5	48.5	25.2	
V100 PCIe (16GB)	2.5	1,100.7	424.1	125.2	34.1	17.2	
Quadro RTX 8000	2.5	1,317.9	420.9	110.2	31.1	16.2	2.9
Quadro RTX A6000	2.5	**1,704.9**	571.6	162.3	46.7	24.2	4.3
A100 SXM4 (40GB)	2.5	1,455.8	569.8	**198.3**	**63.5**	**31.2**	**5.7**

Table 4. Benchmark results for simulations run with a 4.0-fs time step. Simulation rates are reported in simulated nanoseconds per day. Missing entries indicate that a benchmark did not run because it exceeded the memory capacity of the GPU.

GPU	Time step (fs)	DHFR	ApoA1	F1-ATPase	STMV	Ribosome	10× STMV
GeForce GTX 1080 Ti	4.0	1,260.8	342.4	68.7	22.3	12.1	
GeForce RTX 2080	4.0	1,588.9	462.7	99.0	29.8		
GeForce RTX 2080 Ti	4.0	1,856.9	607.1	142.6	40.7		
GeForce RTX 3080	4.0	2,125.4	761.8	200.6	59.2		
GeForce RTX 3090	4.0	**2,395.5**	**837.2**	236.8	68.9	35.8	
V100 PCIe (16GB)	4.0	1,546.7	611.3	176.8	48.0	24.4	
Quadro RTX 8000	4.0	1,931.9	623.0	155.0	43.6	23.1	4.0
Quadro RTX A6000	4.0	2,376.3	830.9	231.2	66.5	34.6	6.0
A100 SXM4 (40GB)	4.0	2,027.7	827.6	**288.9**	**92.6**	**45.4**	**8.3**

References

1. (2021, Mar.) Ambergpu benchmarks. [Online]. Available: https://ambermd.org/gpus16/benchmarks.htm

2. (2021, Mar.) NAMD Utilities. [Online]. Available: http://www.ks.uiuc.edu/Research/namd/utilities

3. Lars V. Bock, Christian Blau, Gunnar F. Schröder, Iakov I. Davydov, Niels Fischer, Holger Stark, Marina V. Rodnina, Andrea C. Vaiana, and Helmut Grubmüller, "Energy barriers and driving forces in tRNA translocation through the ribosome," *Nature Structural & Molecular Biology*, 2013, vol. 20, no. 12, pp. 1390–1396.

4. (2021, Mar.) *E. coli* 70S-fMetVal-tRNAVal post-translocation complex, [Online]. Available: http://dx.doi.org/10.2210/pdb4v7a/pdb.

5. Edmond Chow, Charles A. Rendleman, Kevin J. Bowers, Ron O. Dror, Douglas H. Hughes, Justin Gullingsrud, Federico D. Sacerdoti, and David E. Shaw, "Desmond Performance on a Cluster of Multicore Processors," D. E. Shaw Research Technical Report DESRES/TR--2008-01, July 2008.

6. Cristian Predescu, Adam K. Lerer, Ross A. Lippert, Brian Towles, J. P. Grossman, Robert M. Dirks, and David E. Shaw, "The u-series: A separable decomposition for electrostatics computation with improved accuracy," *The Journal of Chemical Physics*, vol. 152, no. 8, 2020, pp. 084113:1–12.

7. Cristian Predescu, Michael Bergdorf, and David E. Shaw, "Midtown Splines: An Optimal Charge Assignment for Electrostatics Calculations*," The Journal of Chemical Physics*, vol. 153, no. 22, 2020, pp. 224117:1–180.

8. K. Anton Feenstra, Berk Hess, and Herman J. C. Berendsen, "Improving efficiency of large timescale molecular dynamics simulations of hydrogen-rich systems," *Journal of Computational Chemistry*, vol. 20, 1999, pp. 786–798.

9. Michael Bergdorf, Eric T. Kim, Charles A. Rendleman, and David E. Shaw, “Desmond/GPU Performance as of November 2014” D. E. Shaw Research Technical Report DESRES/TR—2014-01, November 2014.

10. Michael Bergdorf, Sean Baxter, Charles A. Rendleman, and David E. Shaw, "Desmond/GPU Performance as of November 2016," D. E. Shaw Research Technical Report DESRES/TR--2016-01, November 2016.

Appendix

Supplementary files

Our website hosts the structure[3] and configuration[4] files used for the simulations reported in this document. The structure files for benchmarks using HMR were created with the msys[5] dms-hmr utility.

Desmond/GPU memory requirements

The memory requirement of Desmond/GPU (in MB) can be estimated as:

$$405 + (2.3 \times \text{the number of particles in thousands}).$$

A system with four million atoms will thus require a GPU with at least 9605 MB of RAM.

[3] https://www.deshawresearch.com/downloads/download_trajectory_benchmark2021.cgi
[4] https://www.deshawresearch.com/publications/Desmond-GPU-performance-1.9.1-April-2021.tgz
[5] (2021, Mar.) Msys Library. [Online]. Available: https://github.com/DEShawResearch/msys

Desmond/GPU Availability

Desmond/GPU is available without cost from D. E. Shaw Research[6] for non-commercial research use by non-profit institutions, and under commercial license from Schrödinger, LLC[7] for other purposes or parties.

Acknowledgements

We thank Clayton Falzone, Mark Heily, Sanjeev Pandey, Goran Pocina, and Michael Fenn for setting up our GPU infrastructure; and Eric Martens for editorial assistance.

[6] http://www.deshawresearch.com/resources_desmond html
[7] http://www.schrodinger.com/Desmond