NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

HPC-Café: GROMACS 2024 usage and performance on modern CPUs and GPUs

Dr. Anna Kahler

FAST. FLEXIBLE. FREE.

GROMACS

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

HPC-Café: GROMACS 2024 usage and performance on modern CPUs and GPUs

- MD benchmark systems
- GROMACS on CPU
- GROMACS on GPU
- GROMACS support cases

MD benchmark systems

	Description	# of Atoms	# of Water Molecules	Origin	Special Characteristics
1	R-143a in hexane	20.248	n.a.	Material Sciences	**very high output rate!**
2	a short RNA piece	31.889	10.492	Biochemistry	AMBER14SB force field with OL refinement for RNA and DNA
3	a protein inside a membrane	80.289	52.488	Life Sciences	Berendsen pressure coupling (not recommended anymore)
4	a protein	170.320	53.976	Biochemistry	**our default benchmark**
5	a protein membrane channel	615.924	144.318	from NVIDIA	Charmm36 force field
6	a huge virus protein	1.066.628	299.855	from GROMACS	common setup

NHR FAU

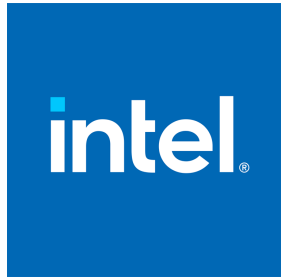
FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

GROMACS on CPU

- CPU systems at NHR@FAU
- GROMACS installation on CPU
- GROMACS usage
- CPU benchmarks with gmx2024

CPU systems at NHR@FAU

- Woody: Intel Xeon Gold 6326, 2 sockets à 16 cores (Ice Lake)
- Fritz: Intel Xeon Platinum 8360Y, 2 socket à 36 cores (Ice Lake)
  Intel Xeon Platinum 8470, 2 sockets à 52 cores (Sapphire Rapid)



- Genoa: AMD EPYC 9654, 2 sockets à 96 cores, SMT opt.
- GenoaX: AMD EPYC 9684X, 2 sockets à 96 cores, SMT opt.
- Bergamo: AMD EPYC 9754, 2 sockets à 128 cores, SMT opt.

- gracehop1: ARM Neoverse V2 (72 cores)

GROMACS installation on CPU

- For Intel CPUs: `spack install`
  - `gromacs@2024%gcc@11.2.0 ~blas ~cuda ~cycle_subcounters ~double +hwloc ~ipo ~lapack ~mdrun_only +mpi ~nosuffix ~opencl +openmp ~plumed ~relaxed_double_precision +shared ~sycl build_type=Release`
- For AMD CPUs based on: https://www.amd.com/en/developer/zen-software-studio/applications/spack/hpc-applications-gromacs.html
  - `spack install gromacs@2024 +openmp build_type=Release %aocc target=zen4 ^amdblis threads=openmp ^amdlibflame ^amdfftw ^openmpi fabrics=auto`
- For ARM CPUs: `spack install gromacs@2024`
  - `REAL MPI: +cuda +mpi [~sve] <^fftw|^armpl-gcc> [^openmpi]`
  - `thread MPI: +cuda ~mpi [<+sve|~sve>] <^armpl-gcc|^fftw ~mpi>`

GROMACS usage

- Performance can be optimized if the number of PME tasks `[-npme]` and openMP threads `[-ntomp]` is specified; different for each simulation.
- SMT can be beneficial but requires testing.

- Non-Intel CPUs: No rank reordering in initializing domain decomposition communicators. Read `hwloc` information from file instead of letting GROMACS do it.

- REAL MPI: **`mpirun -n <# threads> --oversubscribe`** `gmx_mpi -npme <# PME tasks> -ntomp <# openMP threads> -noconfout -nsteps 30000 -resethway -maxh 0.3 -pin on -pinstride 1 -s my.tpr`

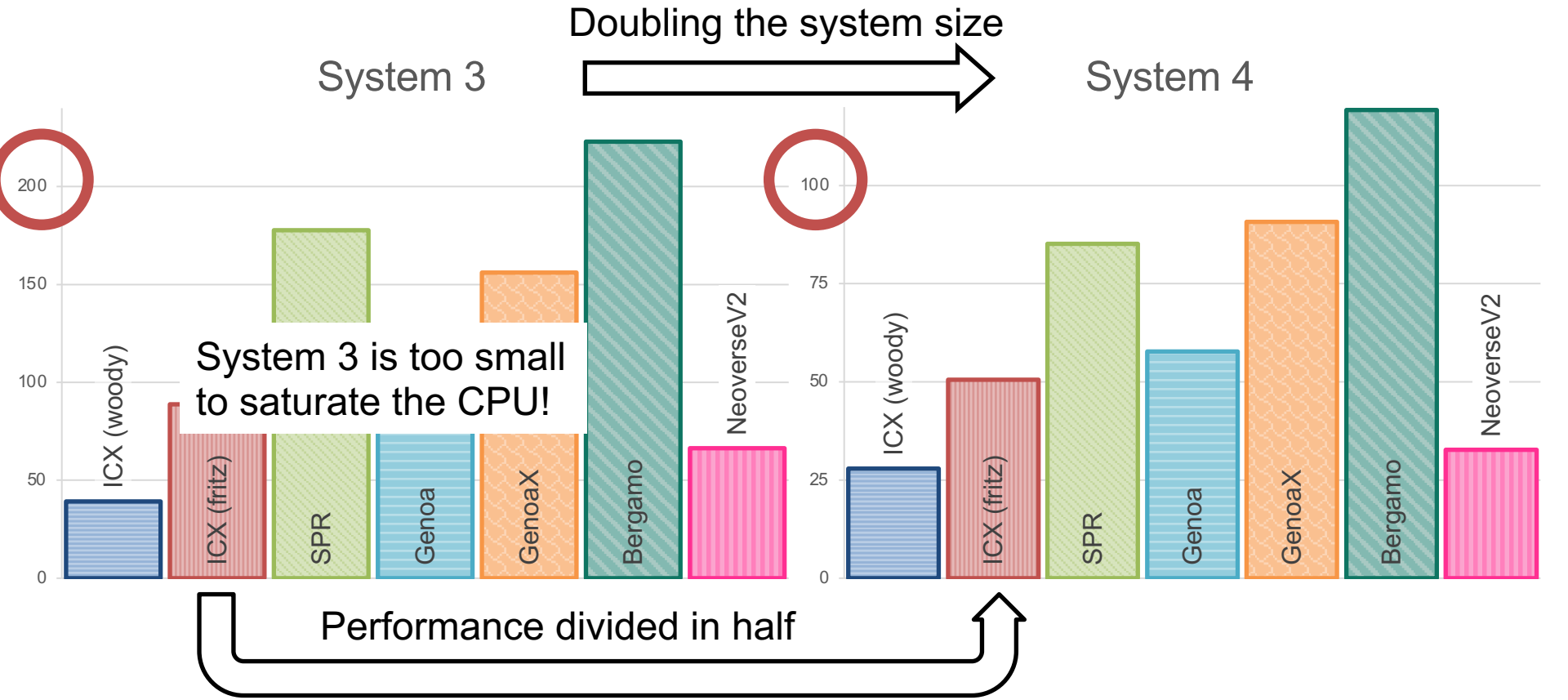
- thread MPI: `gmx -ntmpi <# threads> -npme <# PME tasks> -ntomp <# openMP threads> -noconfout -nsteps 30000 -resethway -maxh 0.3 -pin on -pinstride 1 -s my.tpr`

CPU benchmarks [ns/day], gmx2024

scaling =
frequency x CPU cores

1 node	System 1	System 2	System 3	System 4	System 5	System 6
ICX (woody)	68,2	175,3	39,2	27,9	4,9	n.a.
ICX (Fritz)	127,9	314,9	88,7	50,6	10,5	6,0
SPR	178,1	476,3	177,6	85,2	22,8	12,0
Genoa	101,0	278,3	119,9	57,7	18,6	10,8
GenoaX	140,4	441,0	156,1	90,7	32,0	19,2
Bergamo	193,3	492,7	222,9	119,8	41,0	23,8
Neoverse V2	80,2	193,6	66,4	32,7	7,9	4,4

CPU benchmarks [ns/day], gmx2024



NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

GROMACS on GPU

- NVIDIA GPU systems at NHR@FAU
- GROMACS installation on GPU & usage
- GPU benchmarks with gmx2024
- Performance standardized to system size
- Performance vs hardware costs
- Performance per Watts

NVIDIA GPU systems at NHR@FAU

Name	Architecture	# CUDA cores	Power consumption	RAM	Memory bandwidth
H100	Hopper	16.896	700 W	80 GB	3,35 TB/s
GH200	Hopper	16.896	1,000 W	96 GB	4 TB/s
A100	Ampere	6.912	400 W	40 GB	1,5 TB/s
A40	Ampere	10.752	300 W	48 GB	696 GB/s
L4	Ada Lovelace	7.424	72 W	24 GB	300 GB/s
L40	Ada Lovelace	18.176	300 W	48 GB	864 GB/s
L40S	Ada Lovelace	18.176	350 W	48 GB	864 GB/s

GROMACS installation on GPU & usage

- `spack install gromacs +cuda ~mpi`

- environment variables to improve performance:
  `GMX_GPU_PME_DECOMPOSITION, GMX_USE_GPU_BUFFER_OPS, GMX_DISABLE_GPU_TIMING, GMX_ENABLE_DIRECT_GPU_COMM, GMX_CUDA_GRAPH`

- `gmx mdrun -ntmpi 1 -ntomp 16 -pin on -pinstride 1 -nsteps 200000 -nb gpu -pme gpu -bonded gpu -update gpu -s my.tpr`
  - **`ntmpi`**: Number of thread-MPI threads = number of GPUs
  - **`ntomp`**: Number of OpenMP threads per MPI rank = number of CPU cores

GPU benchmarks [ns/day], gmx2024, ntomp=16

[ns/day]	System 1	System 2	System 3	System 4	System 5	System 6
H100	354,36	1.032,85	400,43	204,96	63,49	37,45
GH200	411,48	1.123,47	470,57	232,61	74,96	44,45
A100	267,48	691,67	265,83	129,09	38,83	23,22
A40	277,01	731,80	268,08	122,27	34,86	19,04
L4	200,86	488,56	169,95	77,76	21,77	11,98
L40	416,12	1.254,67	463,32	212,06	57,76	31,65
L40S	405,33	1.219,72	472,45	229,16	68,28	37,93

GPU benchmarks, gmx2024, ntomp=16

standardized to A40	System 1	System 2	System 3	System 4	System 5	System 6
H100	1,28	1,41	1,49	1,68	1,82	1,97
GH200	1,49	1,54	1,76	1,90	2,15	2,33
A100	0,97	0,95	0,99	1,06	1,11	1,22
A40	1,00	1,00	1,00	1,00	1,00	1,00
L4	0,73	0,67	0,63	0,64	0,62	0,63
L40	1,50	1,71	1,73	1,73	1,66	1,66
L40S	1,46	1,67	1,76	1,87	1,96	1,99

Performance standardized to system size

number of atoms					
System 1	System 2	System 3	System 4	System 5	System 6
20.248	31.889	80.289	170.320	615.924	1.066.628

$(ns/day) * atoms$	average
H100	1,61
GH200	1,86
A100	1,05
A40	1,00
L4	0,65
L40	1,67
L40S	1,79

Performance vs hardware costs

Costs incl VAT	
H100	30.723,42 €
GH200	29.750,00 €
A100	8.644,16 €
A40	5.087,25 €
L4	2.618,00 €
L40	7.168,56 €
L40S	7.259,00 €

$\left(\frac{ns}{day}\right) * atoms / €$	average
H100	0,27
GH200	0,32
A100	0,62
A40	1,00
L4	1,27
L40	1,18
L40S	1,25

Performance vs hardware costs

Costs incl VAT	
H100	30.723,42 €
GH200	29.750,00 €
A100	8.644,16 €
A40	5.087,25 €
L4	2.618,00 €
L40	7.168,56 €
L40S	7.259,00 €

[ns/day]	System 1	System 2	System 3	System 4	System 5	System 6
GH200	411,48	1.123,47	470,57	232,61	74,96	44,45
L40	416,12	1.254,67	463,32	212,06	57,76	31,65

Performance per Watts

```
nvidia-smi -q -i $CUDA_VISIBLE_DEVICES -d POWER,UTILIZATION --loop
```

power draw average [W]

	System 1	System 2
H100		
GH200		
A100		
A40		
L4		
L40		
L40S		

	$\left(\frac{ns}{day}\right) * atoms/W$
H100	
GH200	
A100	
A40	
L4	
L40	
L40S	

$\left(\frac{ns}{day}\right) * atoms/W$	average
H100	1,26
GH200	1,44
A100	1,24
A40	1,00
L4	2,63
L40	1,72
L40S	1,75

GROMACS support cases

- Retaining performance & reducing hardware costs
- Twice the performance on half of the resources
- Performance gain & no changes to the original compilation

Retaining performance & reducing hardware costs!

- Type of Simulation: REMD (26 replicas) with GROMACS
- Initial Performance: **124 ns/day** on 12 dual-socket Intel IceLake nodes
- Estimated Hardware costs: **€ 60,800**

- Solution: run REMD simulation on GPUs
- Problem: number of replicas is not a multiple of eight
  - assign PP- and PME-tasks to the GPUs by hand
  - `-gputasks 0123456701234567012345670123456701234567012345670123`

- Final Performance: **120.6 ns/day** on eight Nvidia A40 GPUs
- Estimated Hardware costs: **€ 48,700**

Twice the performance on half of the resources!

- Type of Simulation: MD with GROMACS but 2,600,000 atoms
- Initial Performance: **11.8 ns/day** on eight NVIDIA A40 GPUs

- Solution: choose appropriate environment variables & adjust runtime parameters to the hardware
  - for improved communication between GPUs:

```
export GMX_GPU_PME_DECOMPOSITION=1
export GMX_USE_GPU_BUFFER_OPS=1
export GMX_DISABLE_GPU_TIMING=1
export GMX_ENABLE_DIRECT_GPU_COMM=1
```

  - -ntmpi = a multiple of #GPUs
  - -ntomp = product of ntmpi×ntomp = number of cores

- Final Performance: **20 ns/day** on four NVIDIA A40 GPUs

Performance gain & no changes to the original compilation!

- Problem: run GROMACS on 96-core dual-socket AMD Zen4 “Genoa”
  (= 192 cores in the node)
- Compilation: with AMD toolchain using Spack
- Initial Performance: **36 ns/day**

- Surprise: Intel-based binary, compiled on a different hardware → **80 ns/day**

- Analyses: a large number of MPI threads for GROMACS
  - provide topology information: run the `hwloc` command `lstopo` &
    `export HWLOC XMLFILE=<path-to-file>`
  - consider initial warmup phase of the CPU

- Final performance: **100 ns/day**

NHR FAU

FAU Friedrich-Alexander-Universität Erlangen-Nürnberg

- If in doubt, run benchmarks.
- Don’t be embarrassed if you need help.
- Get in touch with your support.

hpc-support@fau.de