Hitchhiker's guide
from traditional HPC cluster to containerized ensemble run at scale

How to lift GROMACS into a cloudy SLURM cluster and evolve to run GROMACS globally using containers

Carsten Kutzner,
Vytautas Gapsys

Max Planck Institute for Biophysical Chemistry
Theoretical and Computational Biophysics
Göttingen

Christian Kniep

Amazon Web Services
Amazon Development Center Germany
Berlin

a few words about where I come from ...

- I work @ Max Planck Institute for Biophysical Chemistry in Göttingen

- Department of Theoretical and Computational Biophysics

- We are mainly doing biomolecular numerical simulations of proteins, membranes, membrane channels, and larger biological "nano-machines" as e.g. ribosomes, in atomic detail

- For these biomolecular simulations we are using the open source software package GROMACS to which we also contribute

Carsten Kutzner

Twitter:

@kutznercarsten https://twitter.com/kutznercarsten

@CompBioPhys https://twitter.com/CompBioPhys

Homepage:

https://www.mpibpc.mpg.de/grubmueller/kutzner

Outline

Introduction:

- biomolecular simulations and their challenges
- the GROMACS software suite

} our exemplary use case

GROMACS in the cloud for

a) High performance computing (HPC)

b) High throughput computing (HTC)

User experience: Setting up a cloud-based HPC cluster with Spack and ParallelCluster

Results

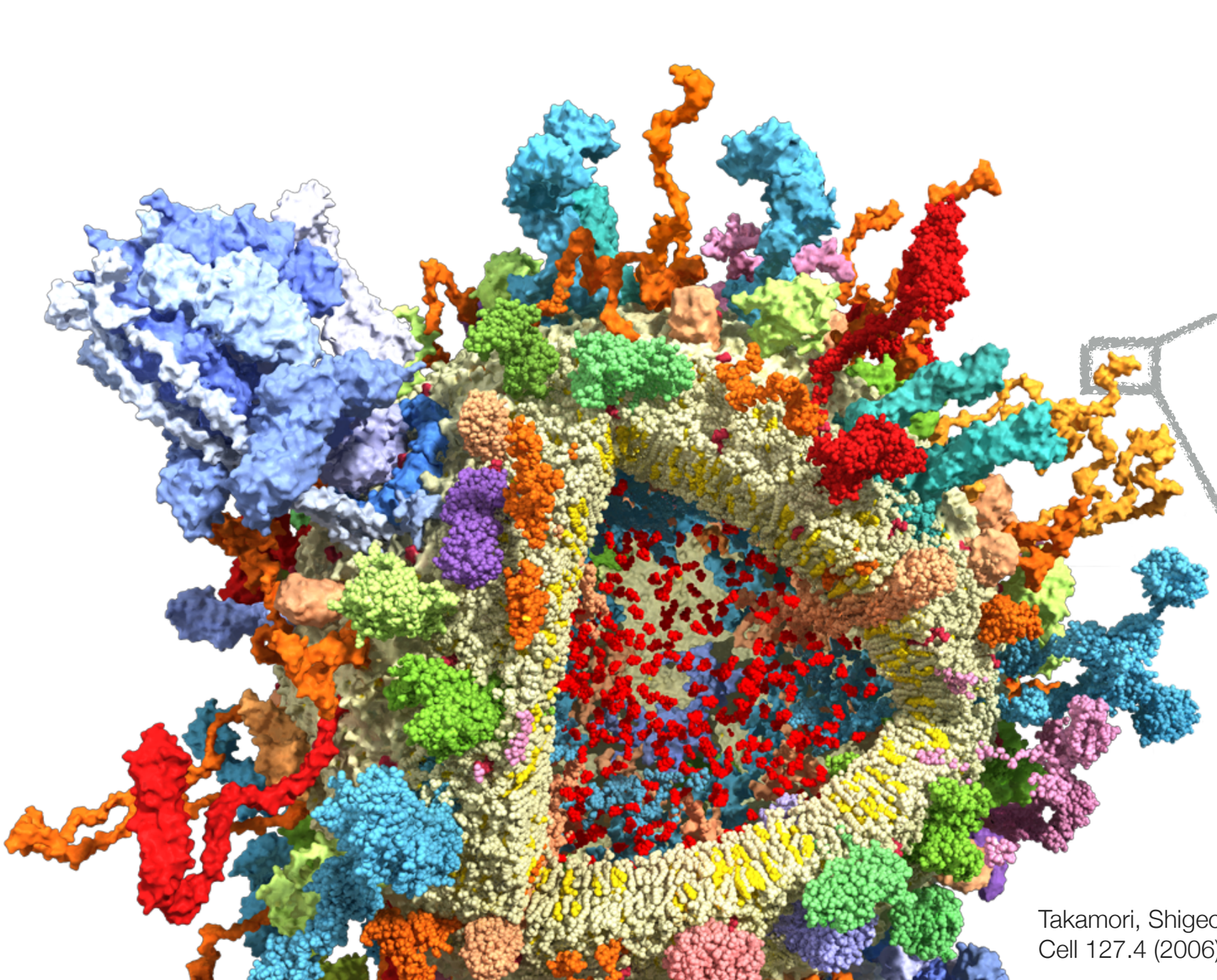
a) HPC on a cloud-based cluster: GROMACS performance benchmarks

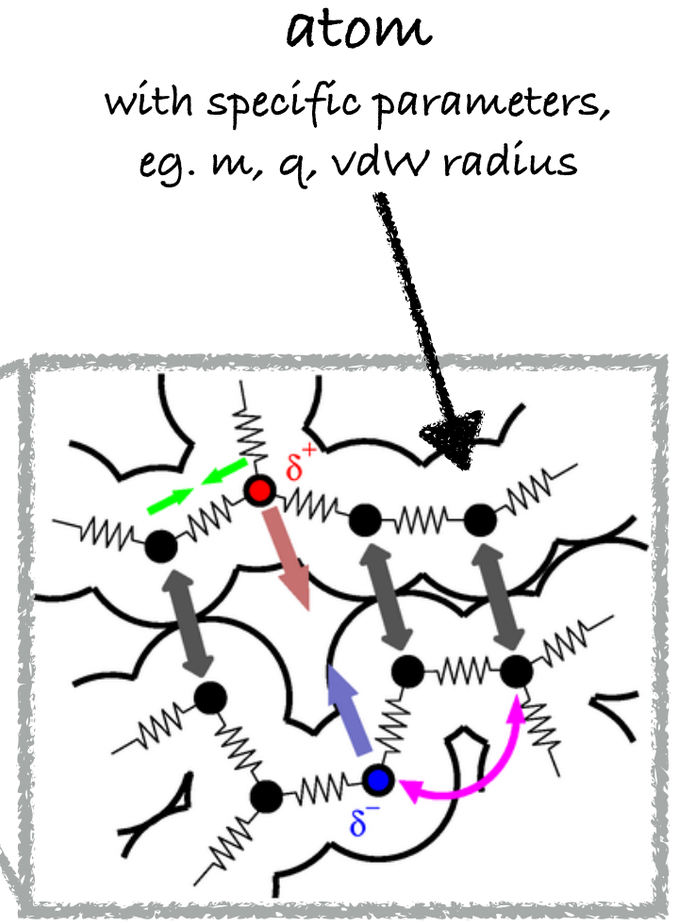
Work in progress

b) HTC in the cloud. Reducing the time-to-solution with massive parallelism to speed up computational drug design

Molecular dynamics simulations

- solve Newton's eq. of motion for N_{at} atoms (N_{at} = 10,000, ... 10,000,000)
- positions **r**i, charges qi, masses mi
- calculate forces from a "force field" potential U(**r**i, qi, mi, ...)





Takamori, Shigeo, et al. *Molecular anatomy of a trafficking organelle.* Cell 127.4 (2006): 831-846.

Molecular dynamics simulations with GROMACS

- GROMACS: open source molecular dynamics (MD) package www.gromacs.org
- currently ~30 active developers from around the globe (Stockholm, Uppsala, US, ...)
- written in C++, with GPU support via CUDA or OpenCL, runs on a wide range of hardware
- extremely optimized for simulation performance, domain decomposition with dynamic load balancing, multi-level parallelism
  - SIMD
  - OpenMP threads
  - MPI ranks
  - multiple-program, multiple-data electrostatics
  - GPU offloading

Challenges of molecular dynamics simulations

- to reach biologically relevant time scales we need ~ microsecond long trajectories
  ➔ **millions of time steps** for a single trajectory!

- usually we are not interested in a single trajectory,
  but in the **average behavior** of a system

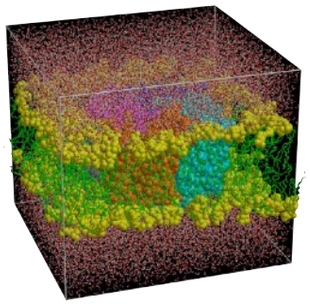
Challenges of molecular dynamics simulations

- to reach biologically relevant time scales we need ~ microsecond long trajectories
  ➔ **millions of time steps** for a single trajectory!

- usually we are not interested in a single trajectory,
  but in the **average behavior** of a system

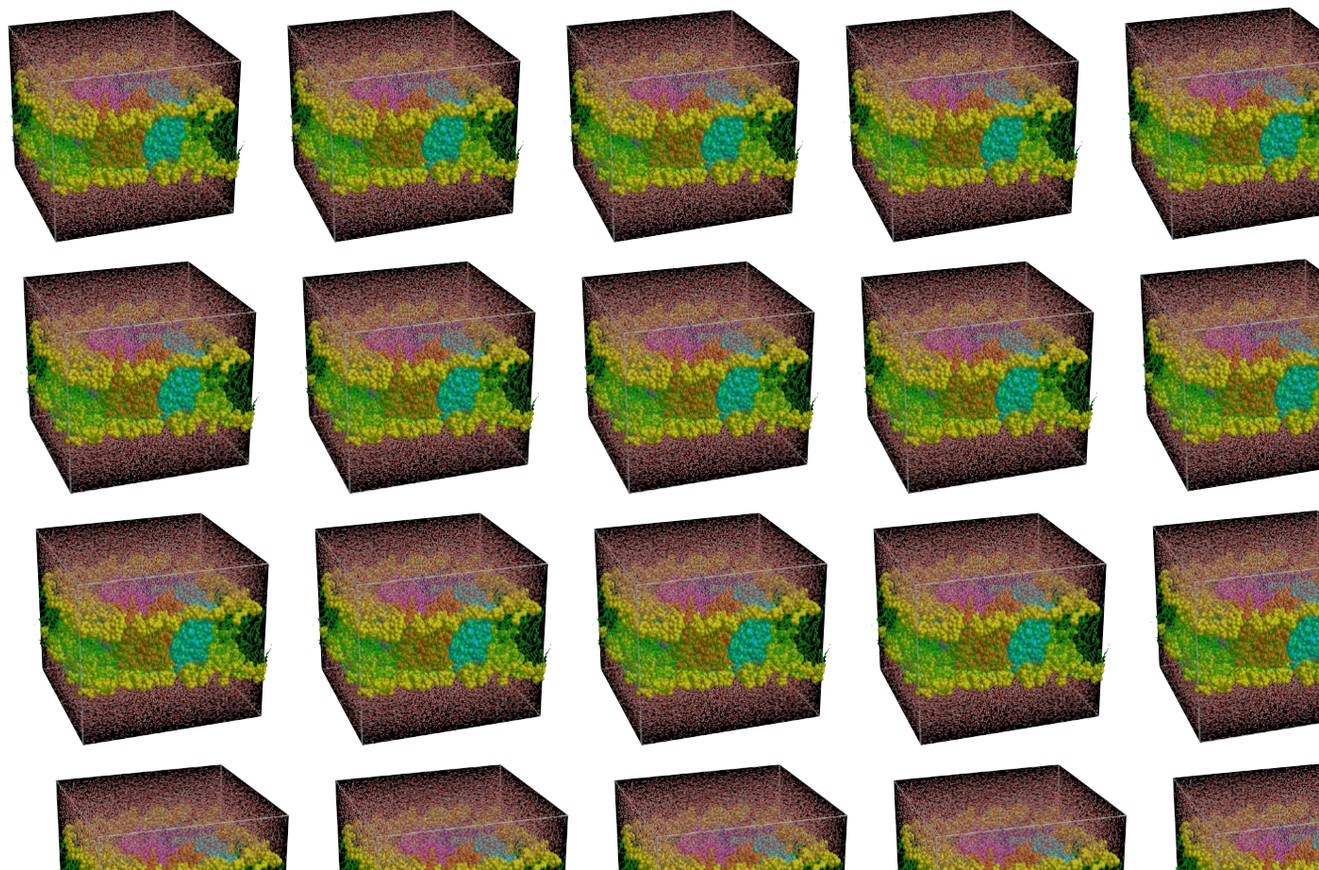
  single simulation ➔ run ensemble of many copies for robust statistics

single simulation



thousands of
similar copies

ensemble of similar simulations



MD simulations: HPC and HTC extreme case scenarios

large system:

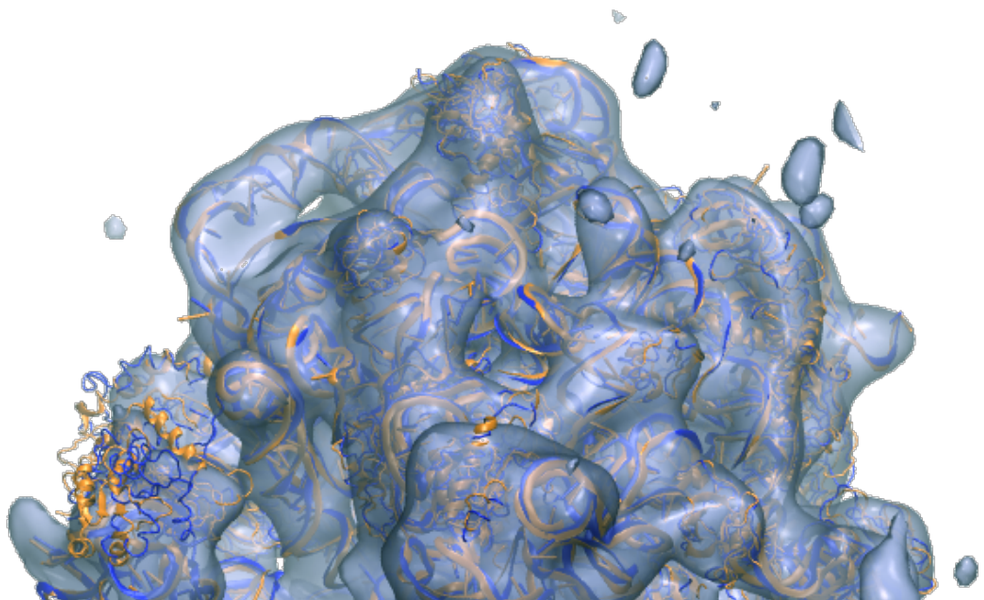
a few long simulations

➔ minimize time-to-solution

HPC

high performance computing

e.g. the ribosome "nanomachine"



Bock, Lars V., et al. *Energy barriers and driving forces in tRNA translocation through the ribosome.* Nature structural & molecular biology 20.12 (2013): 1390-1396.

MD simulations: HPC and HTC extreme case scenarios

large system:

a few long simulations
➔ minimize time-to-solution

HPC

high performance computing

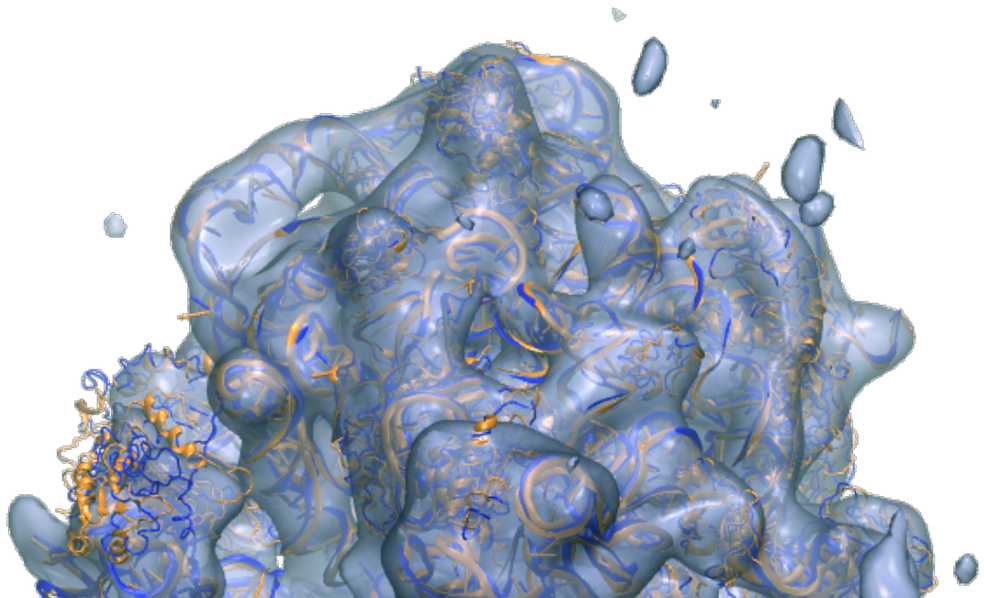
many small systems:

collect as much statistics as possible
➔ maximize throughput

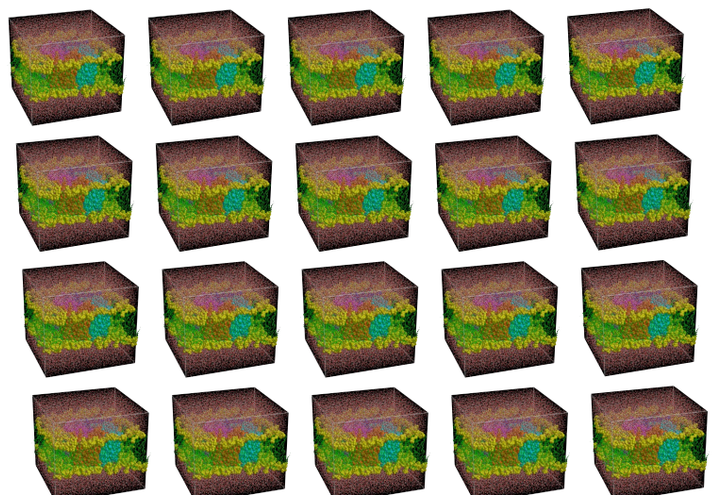
HTC

high throughput computing

e.g. the ribosome "nanomachine"



e.g. ensemble runs for
computational drug design



MD simulations: HPC and HTC extreme case scenarios

large system:

a few long simulations
➔ minimize time-to-solution

many small systems:

collect as much statistics as possible
➔ maximize throughput

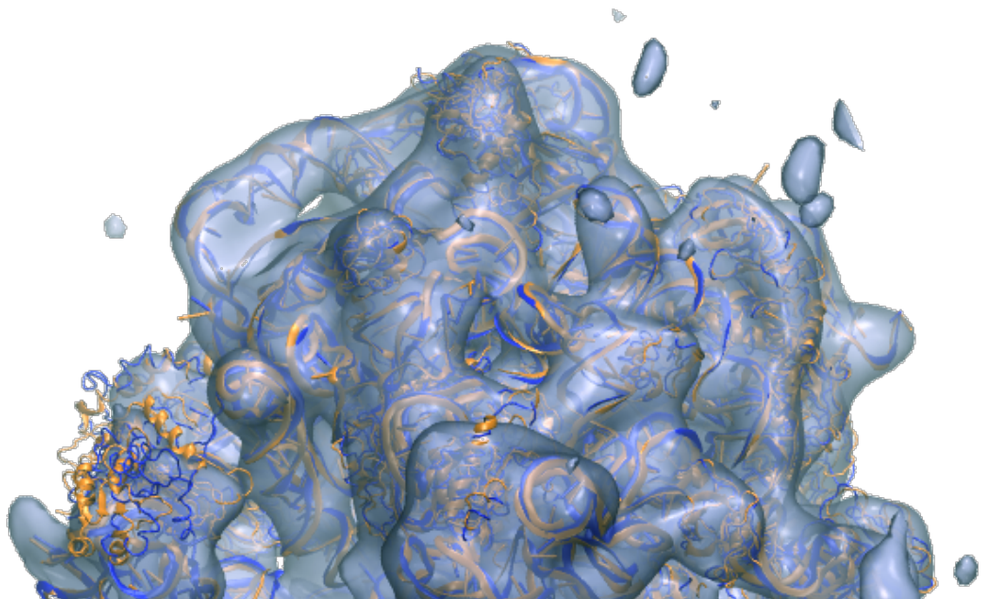
HPC
high performance computing

MD simulations

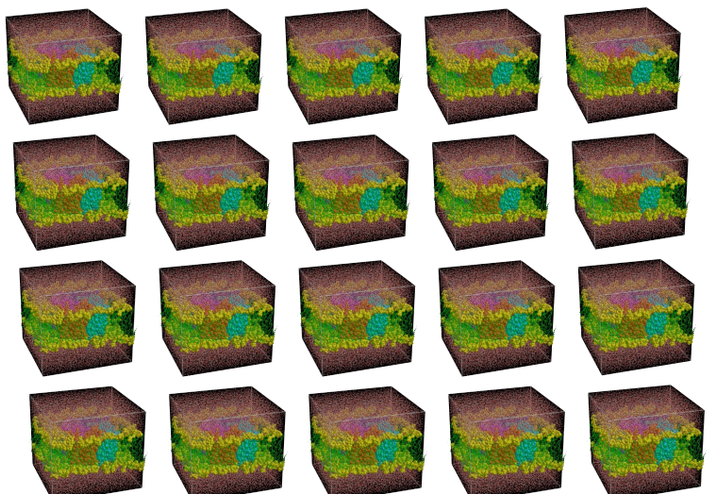
HTC
high throughput computing

- MD can range from HPC to HTC, depending on the questions addressed
- in any case, simulations may take weeks to months!

e.g. the ribosome "nanomachine"



e.g. ensemble runs for computational drug design



Where to run your simulations?

Wherever there is compute time available :)

Sources of compute time for research groups
(research centers / universities / companies):

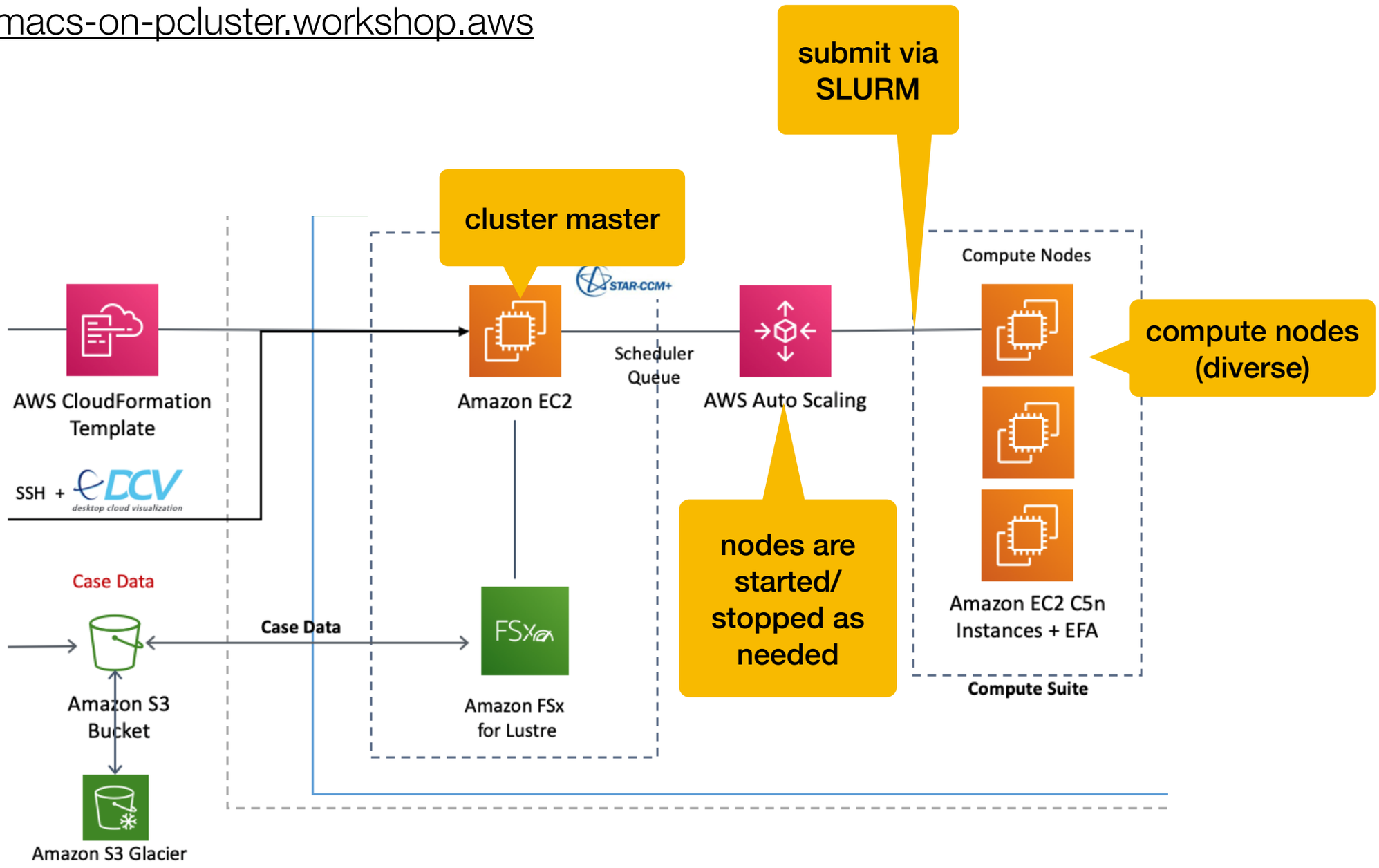
1. **Department cluster.** If space (+cooling infrastructure) permits (often small, long turnover times)
2. **University-wide / company-wide compute cluster**.
   Shared with many others
3. **Supercomputing center.** Nation-wide or European, requires project proposal to go through peer-review, compute time must be used in the allocated time frame.
   They usually very much prefer HPC over HTC! ("expensive interconnects")

4. **In the cloud.** Supports HPC & HTC. Cluster size as needed, various hardware options available, no room, no personal.

Building an cloud-based HPC cluster with ParallelCluster

- https://github.com/aws/aws-parallelcluster
  open source, free, cluster management tool

- Workshop available at
  https://gromacs-on-pcluster.workshop.aws



Installing software on cluster with Spack

- https://github.com/spack/spack.git
- Spack: flexible package manager for supercomputers, Linux, and MacOS
- easily install multiple versions of a software
- workshop available at https://gromacs-on-pcluster.workshop.aws

Installing software on cluster with Spack

- https://github.com/spack/spack.git
- Spack: flexible package manager for supercomputers, Linux, and MacOS
- easily install multiple versions of a software
- workshop available at
  https://gromacs-on-pcluster.workshop.aws

default GROMACS install (2020) with OpenMPI

```
spack install gromacs

 spack install gromacs+cuda

 spack install gromacs@2021.rc1 +cuda ^intel-mpi
```

Installing software on cluster with Spack

- https://github.com/spack/spack.git
- Spack: flexible package manager for supercomputers, Linux, and MacOS
- easily install multiple versions of a software
- workshop available at https://gromacs-on-pcluster.workshop.aws

```
spack install gromacs

spack install gromacs+cuda

spack install gromacs@2021.rc1 +cuda ^intel-mpi
```

with GPU support

Installing software on cluster with Spack

- https://github.com/spack/spack.git
- Spack: flexible package manager for supercomputers, Linux, and MacOS
- easily install multiple versions of a software
- workshop available at https://gromacs-on-pcluster.workshop.aws

```
spack install gromacs

 spack install gromacs+cuda

 spack install gromacs@2021.rc1 +cuda ^intel-mpi
```

GROMACS 2021rc1
with Intel MPI

Know your software ~~or~~ and run benchmarks

- GROMACS is installed on our cluster in the cloud,
  so we directly submit all our jobs?

- ... not yet!

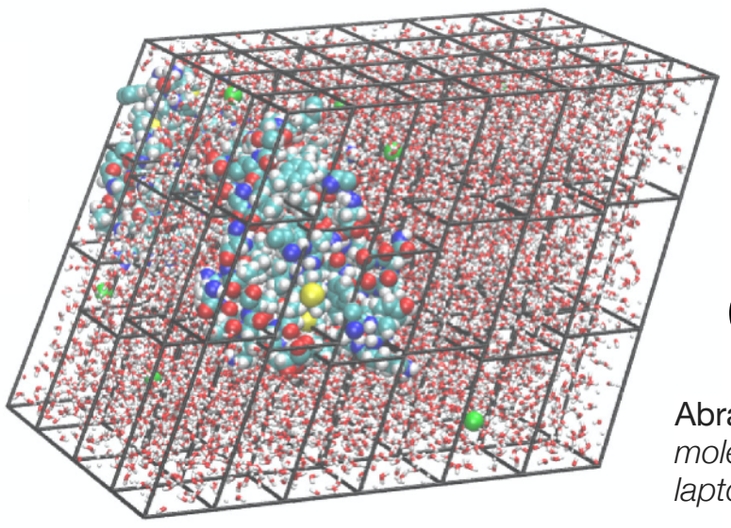
- Simulations may run for weeks – months
  ➔ make sure we don't waste compute time!

- find optimal parallelization parameters with benchmarks

- performance depends on the composition of the input system

- GROMACS has good heuristics
  but does not know about the underlying network performance, e.g.

Know your software ~~or~~ and run benchmarks

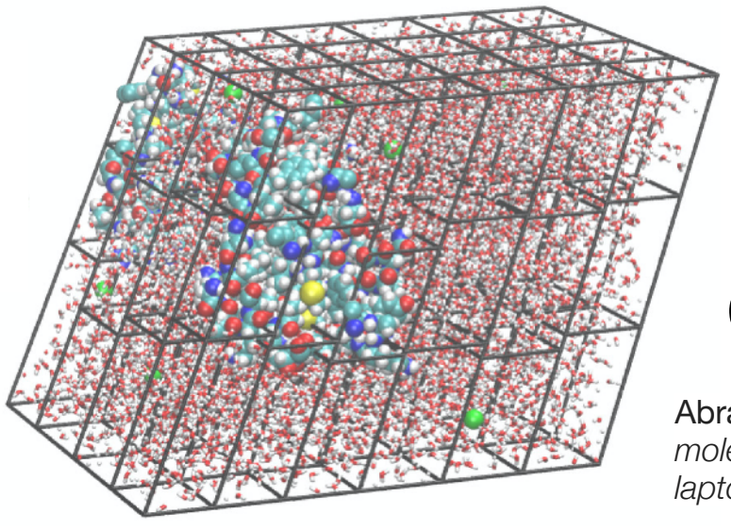


6 x 4 x 3 domains

Abraham, Mark James, et al. *GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers.* ***SoftwareX*** 1 (2015): 19-25.

- **domain decomposition:** system is chopped up into **x** × **y** × **z** domains for parallelization.
  domain = MPI rank
- each domain (MPI rank) ➔ multiple OpenMP threads
- **various possibilities to choose $N_{MPI} \times N_{OpenMP}$ ➔** different performance
- most interactions can be offloaded to GPU(s)
- **all interactions parallelize over multiple CPUs**, but not (yet) over multiple GPUs
  - long-range electrostatic forces do not parallelize over multiple GPUs yet
  - using multiple GPUs efficiently requires a balanced hardware CPU : GPU

Know your software ~~or~~ and run benchmarks



6 x 4 x 3 domains

Abraham, Mark James, et al. *GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers.* ***SoftwareX*** 1 (2015): 19-25.

processor(s) and `instance type`	pricing ($/h)	repli-cas	ranks × threads	PME ranks	MEM (ns/d)
AMD EPYC 7R32	3.696	1	48 × 1	-	67.151
48 cores (96 vCPUs)	3.696	1	96 × 1	-	69.121
3.3 GHz boost	3.696	1	48 × 2	-	71.116
AVX2_128	3.696	1	24 × 3	-	54.175
`c5a.24xlarge`	3.696	1	24 × 4	-	64.981
	3.696	1	16 × 6	-	53.326
	3.696	1	48 × 2	12	67.187
	3.696	1	96 × 1	16	75.047
	3.696	1	96 × 1	24	76.882

>40% difference
in simulation performance

Results a): HPC with GROMACS in the cloud

- Now benchmark GROMACS on various hardware availble in the cloud

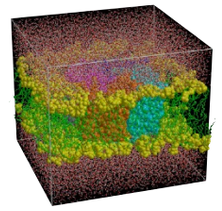
- using MEM and RIB benchmarks

- determine optimal simulation parameters for each setting in terms of $N_{\mathrm{MPI}} \times N_{\mathrm{OpenMP}}$, and others

MEM 81 k atoms

GPU

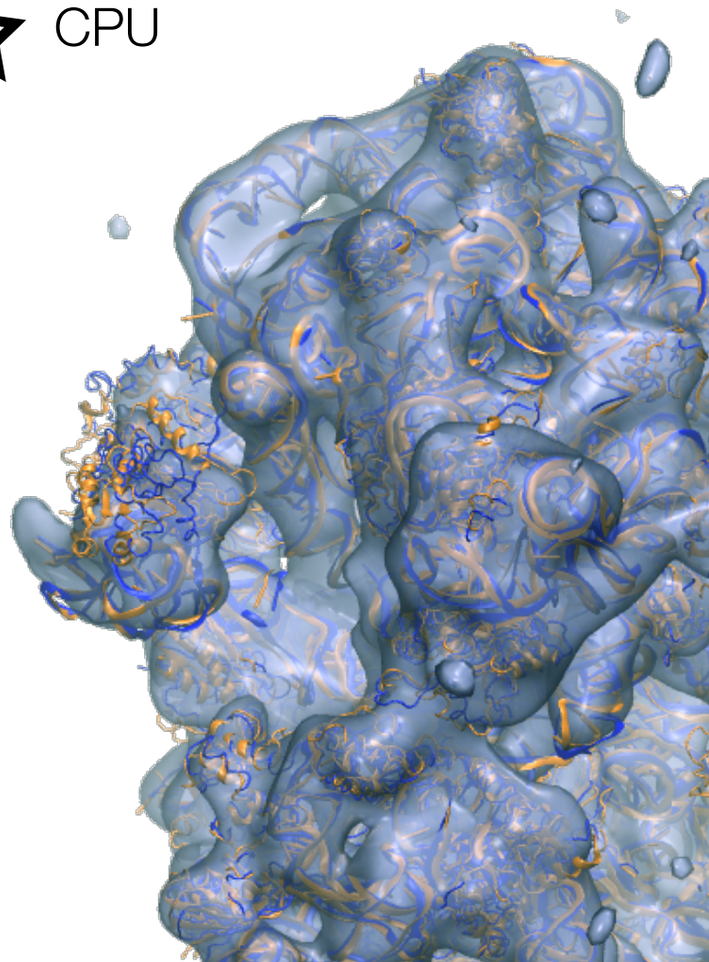
CPU



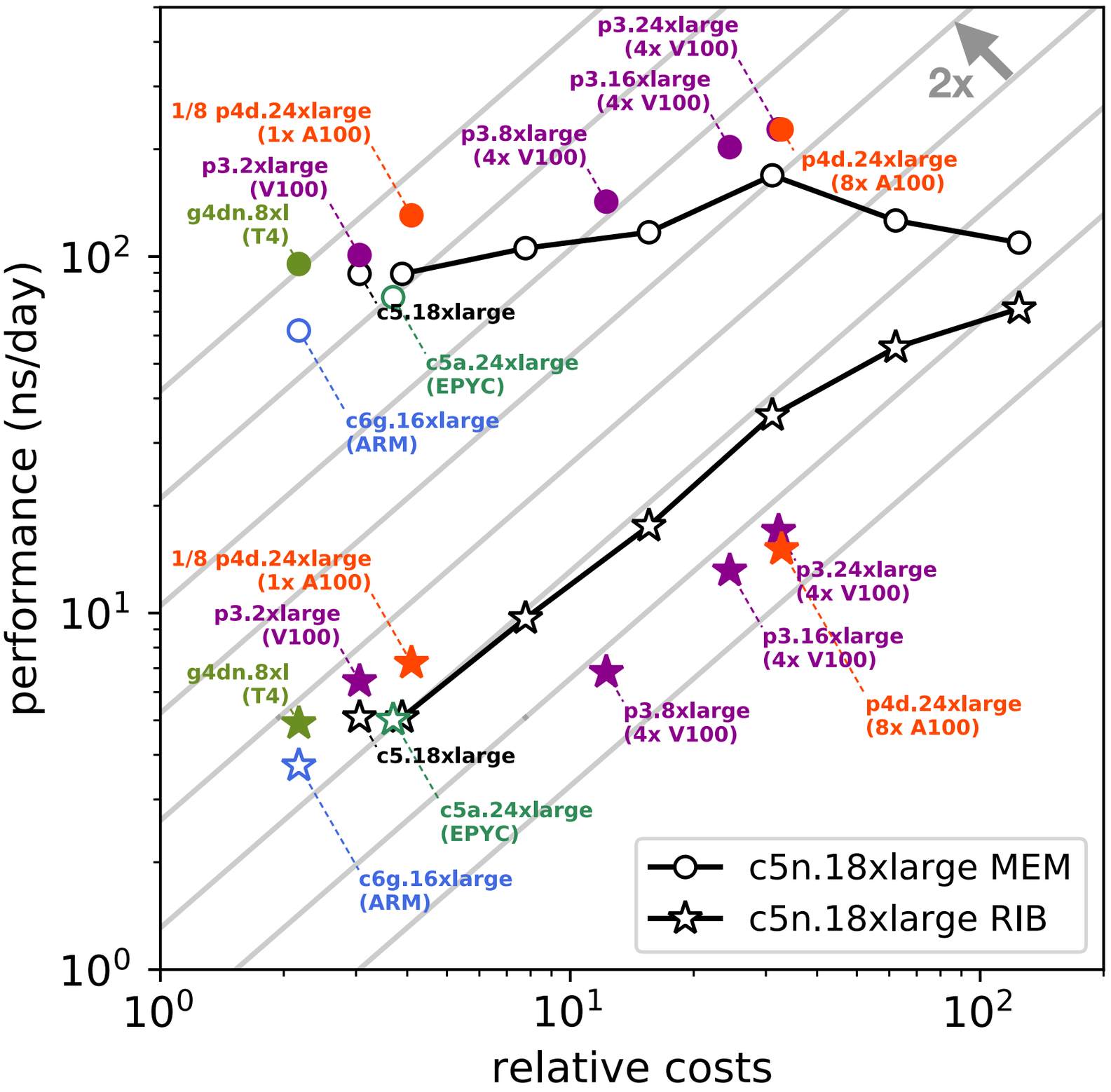
RIB 2 M atoms

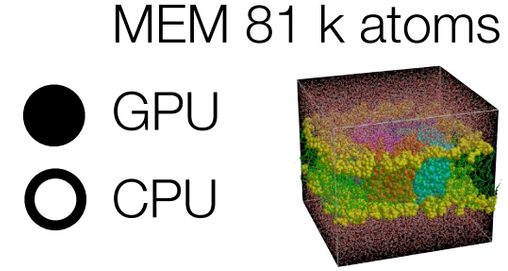
GPU

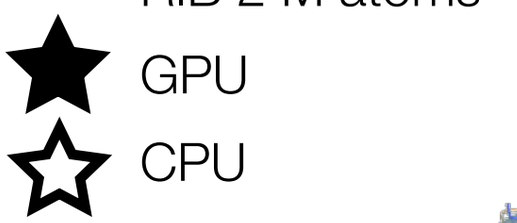
CPU

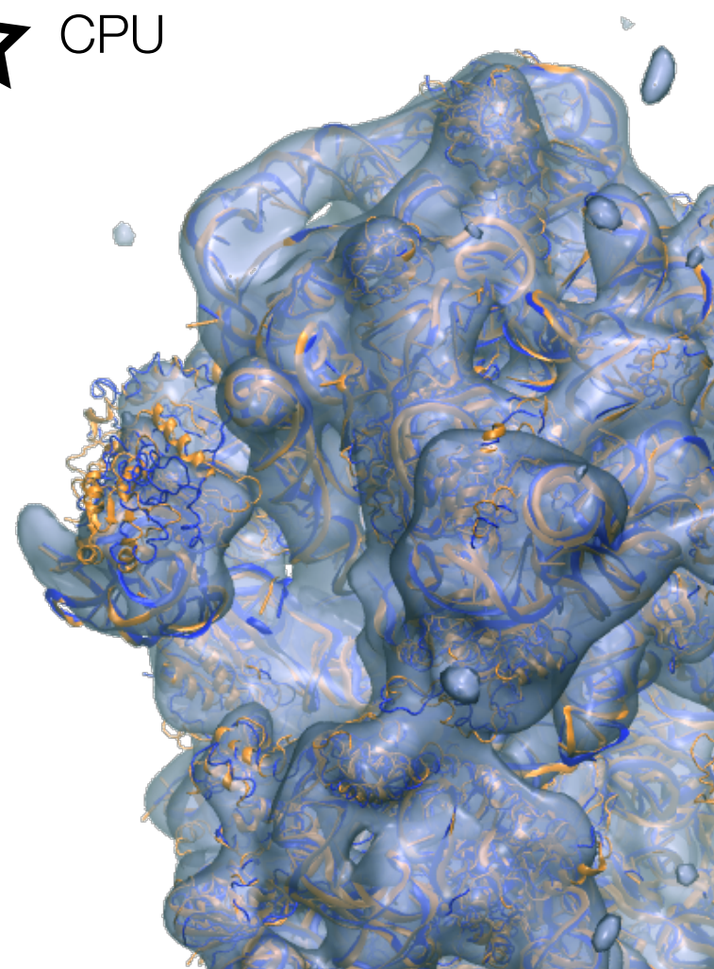


Results a): HPC with GROMACS in the cloud

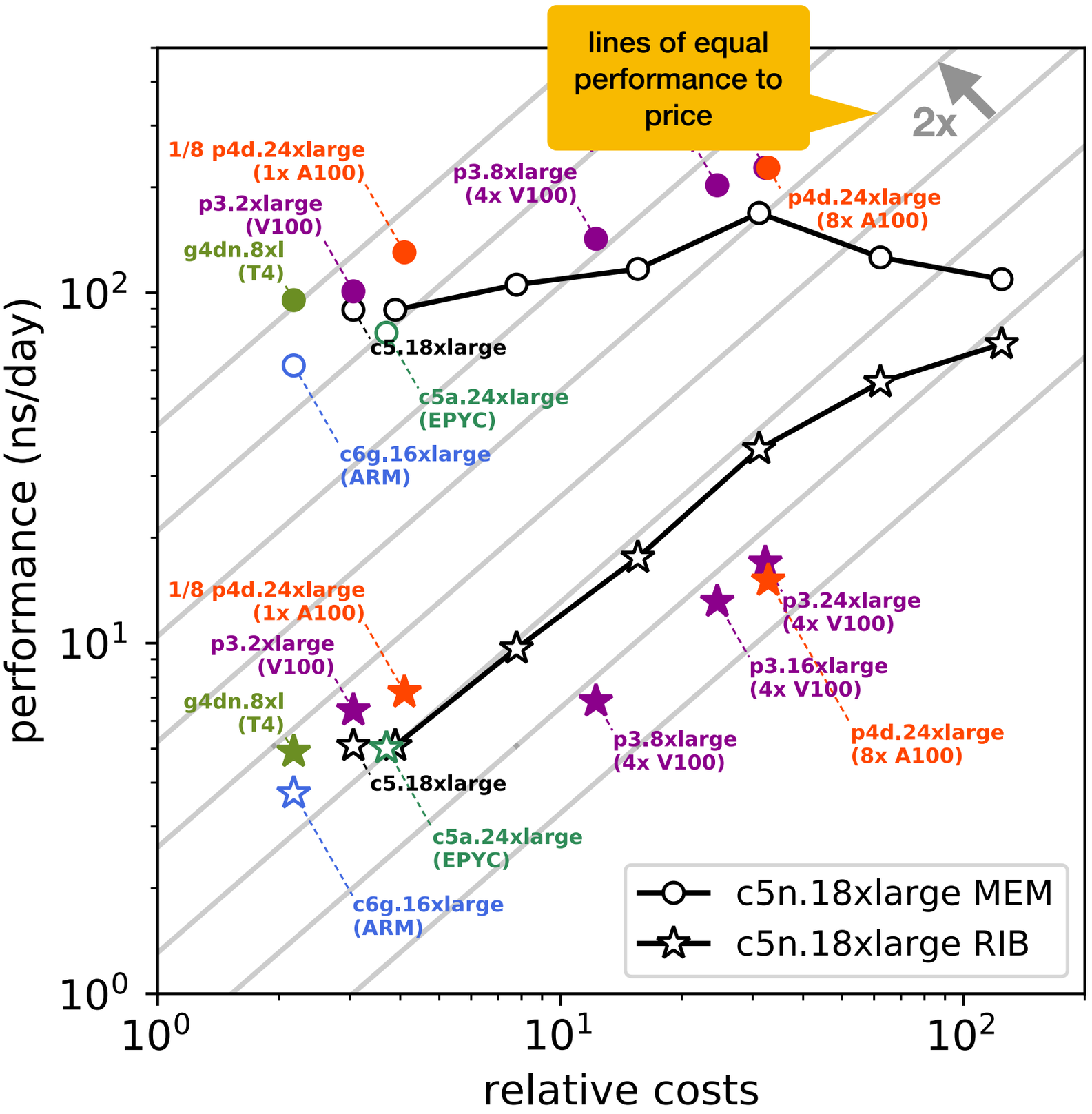








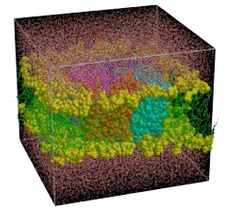
Results a): HPC with GROMACS in the cloud



MEM 81 k atoms

GPU

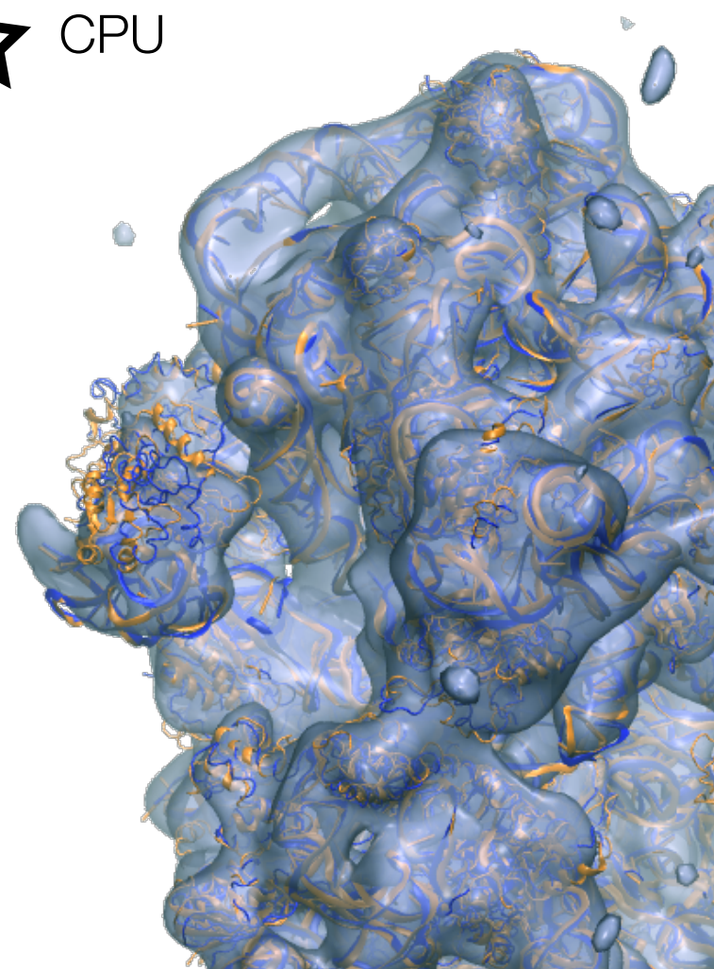
CPU



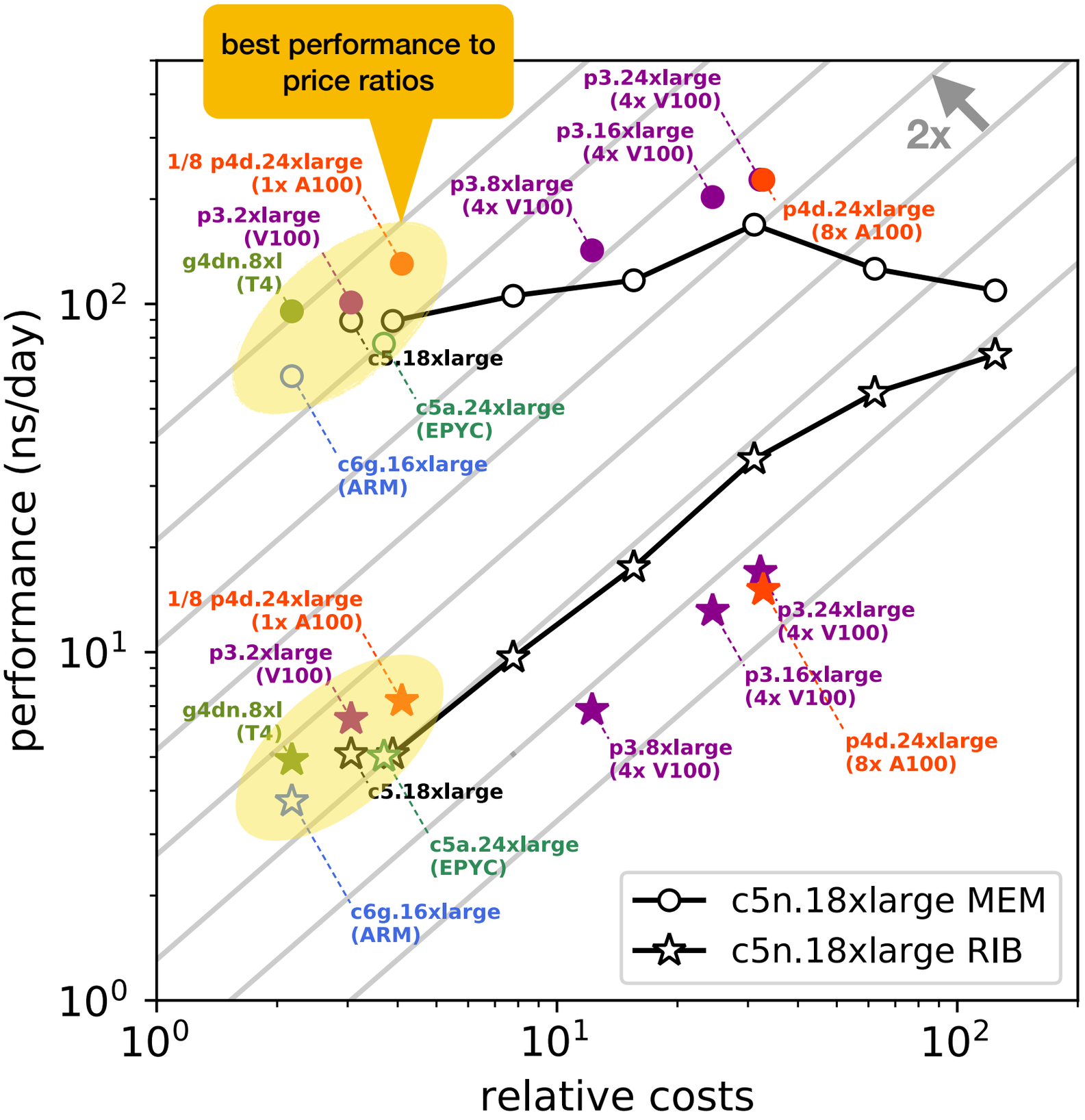
RIB 2 M atoms

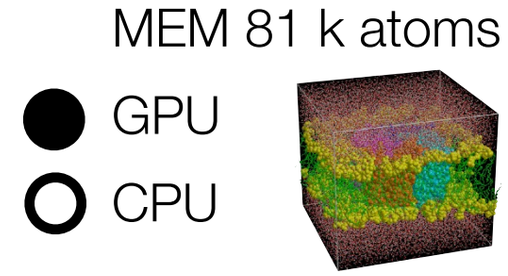
GPU

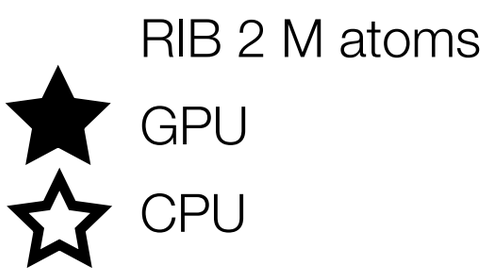
CPU

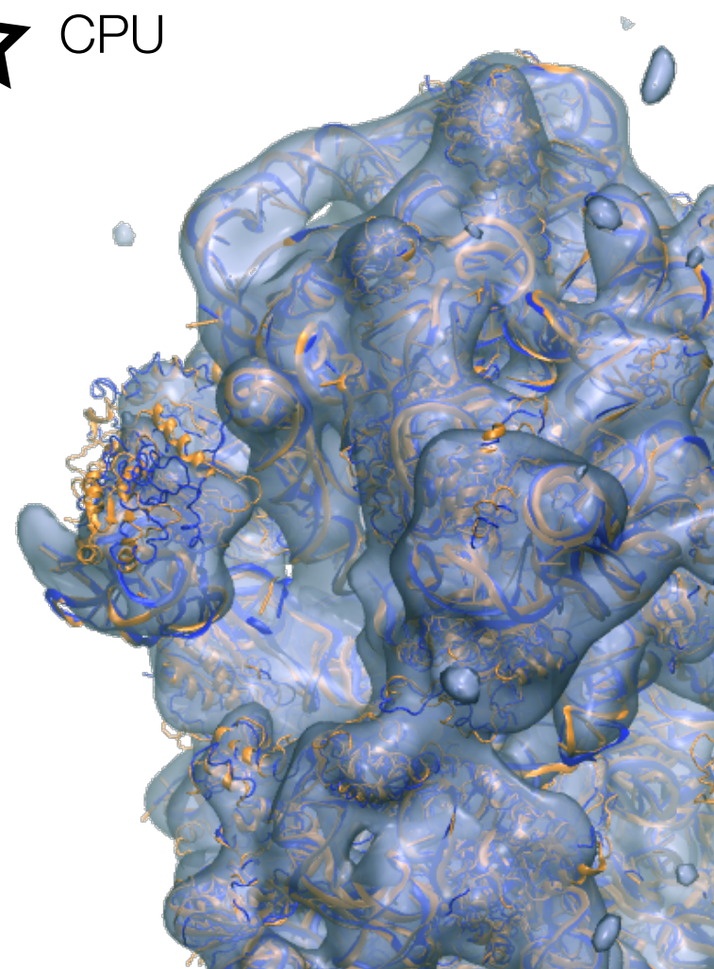


Results a): HPC with GROMACS in the cloud

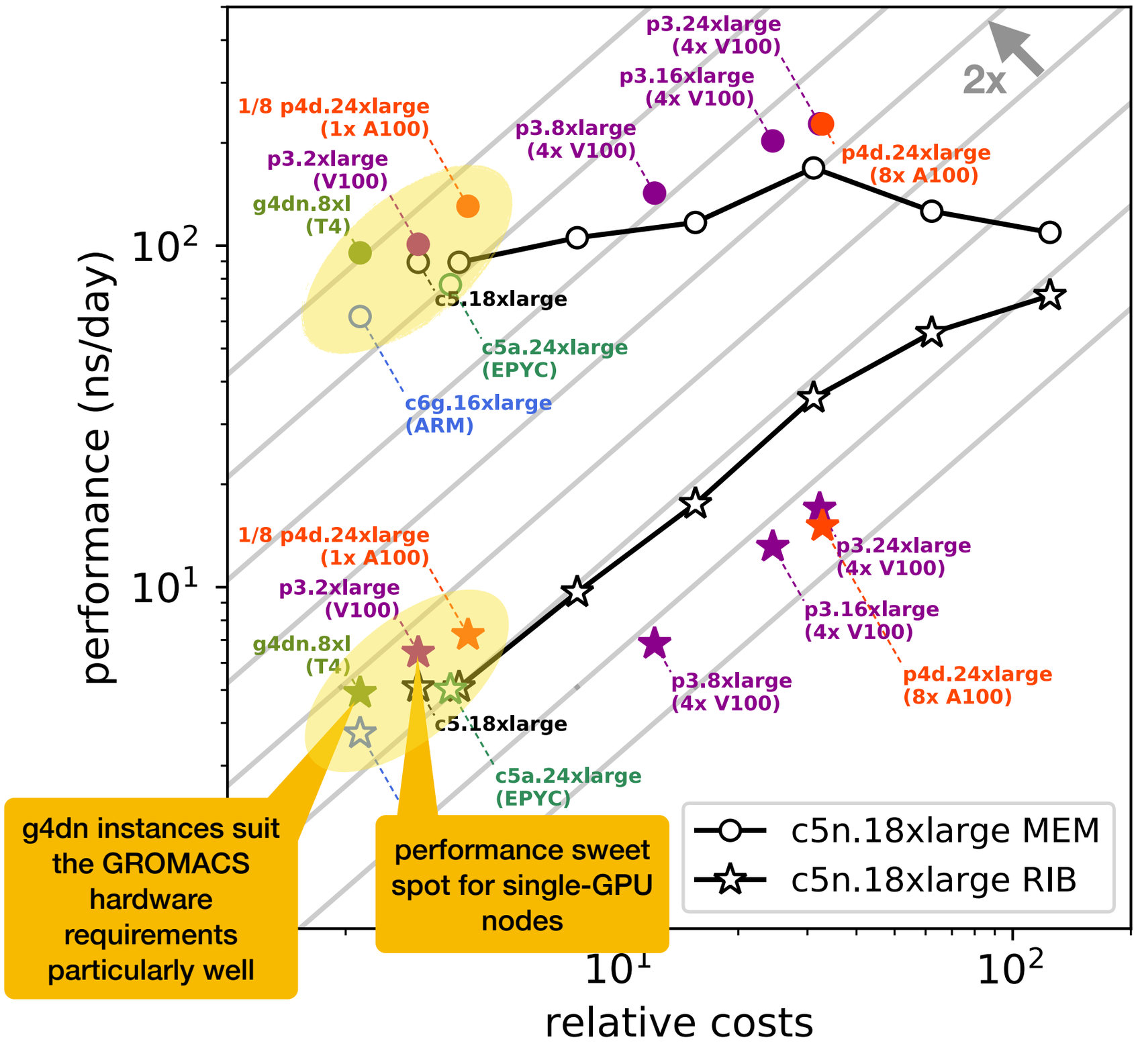








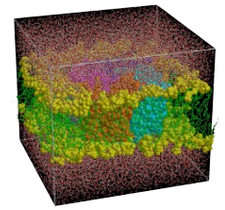
Results a): HPC with GROMACS in the cloud



MEM 81 k atoms

GPU

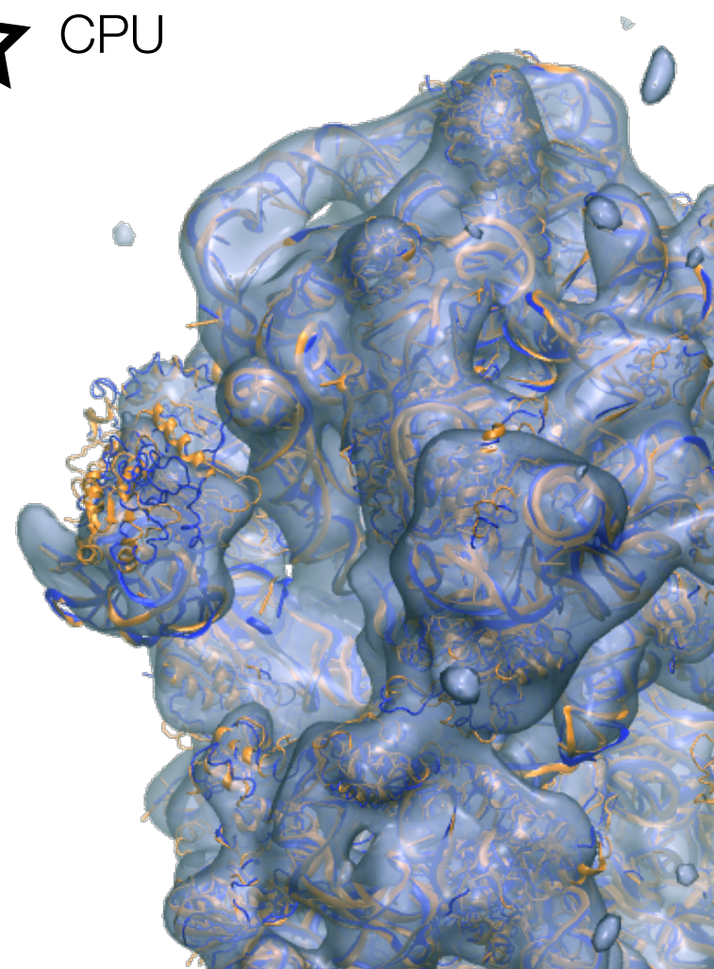
CPU



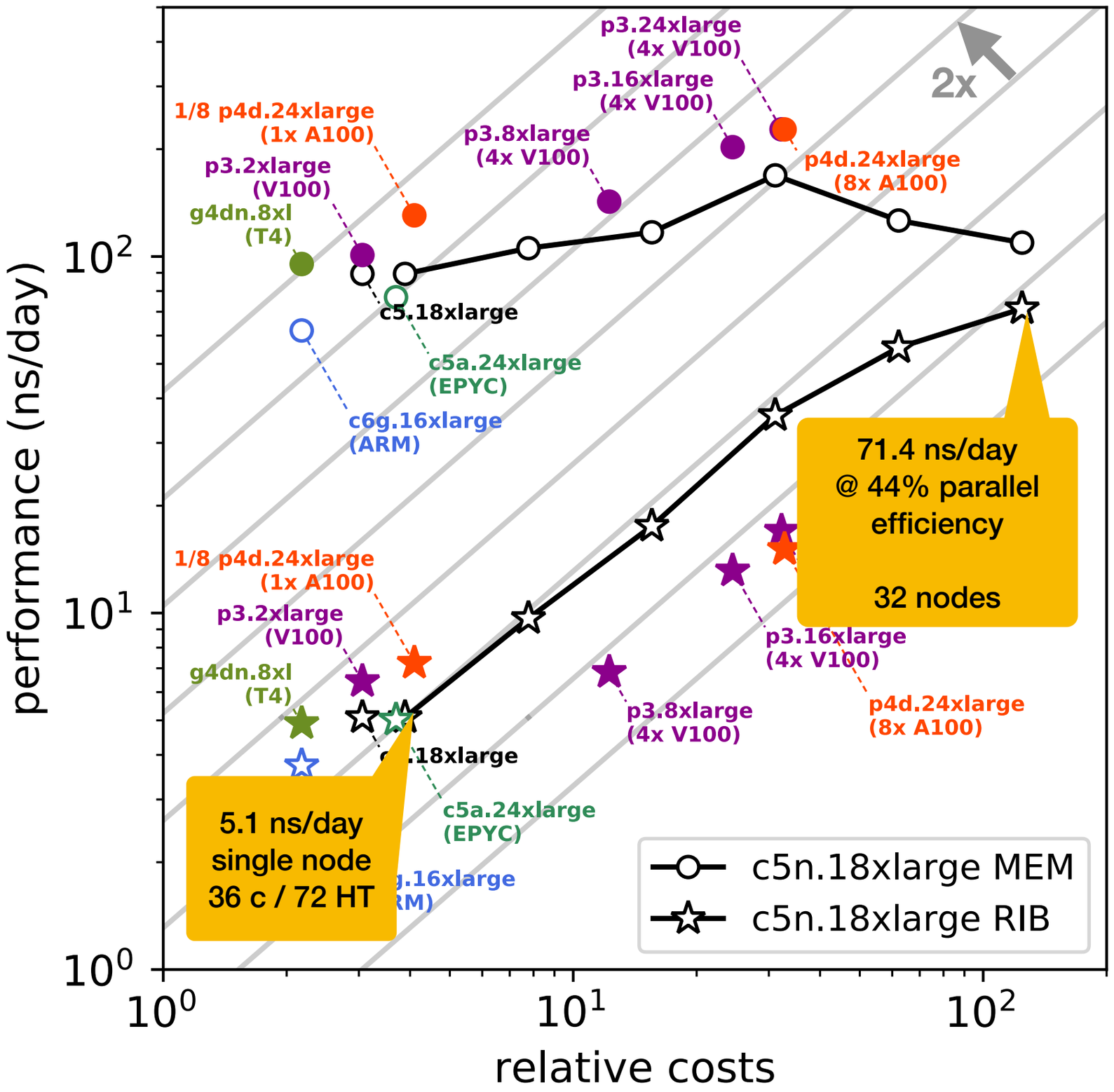
RIB 2 M atoms

GPU

CPU



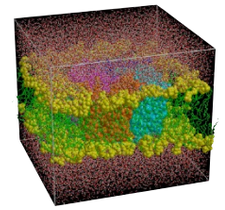
Results a): HPC with GROMACS in the cloud



MEM 81 k atoms

GPU

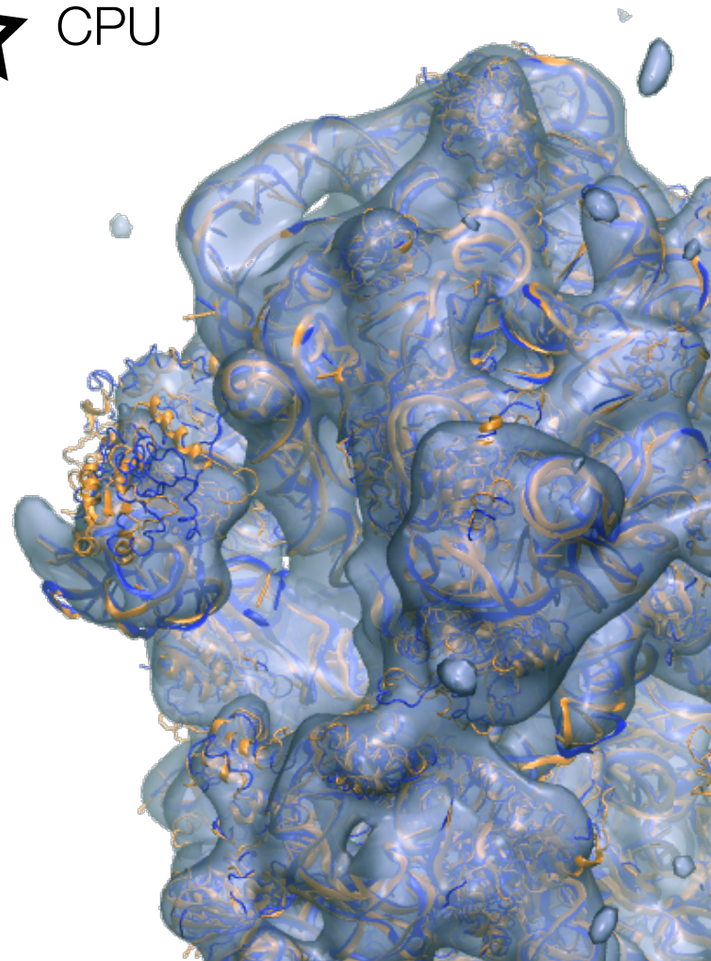
CPU



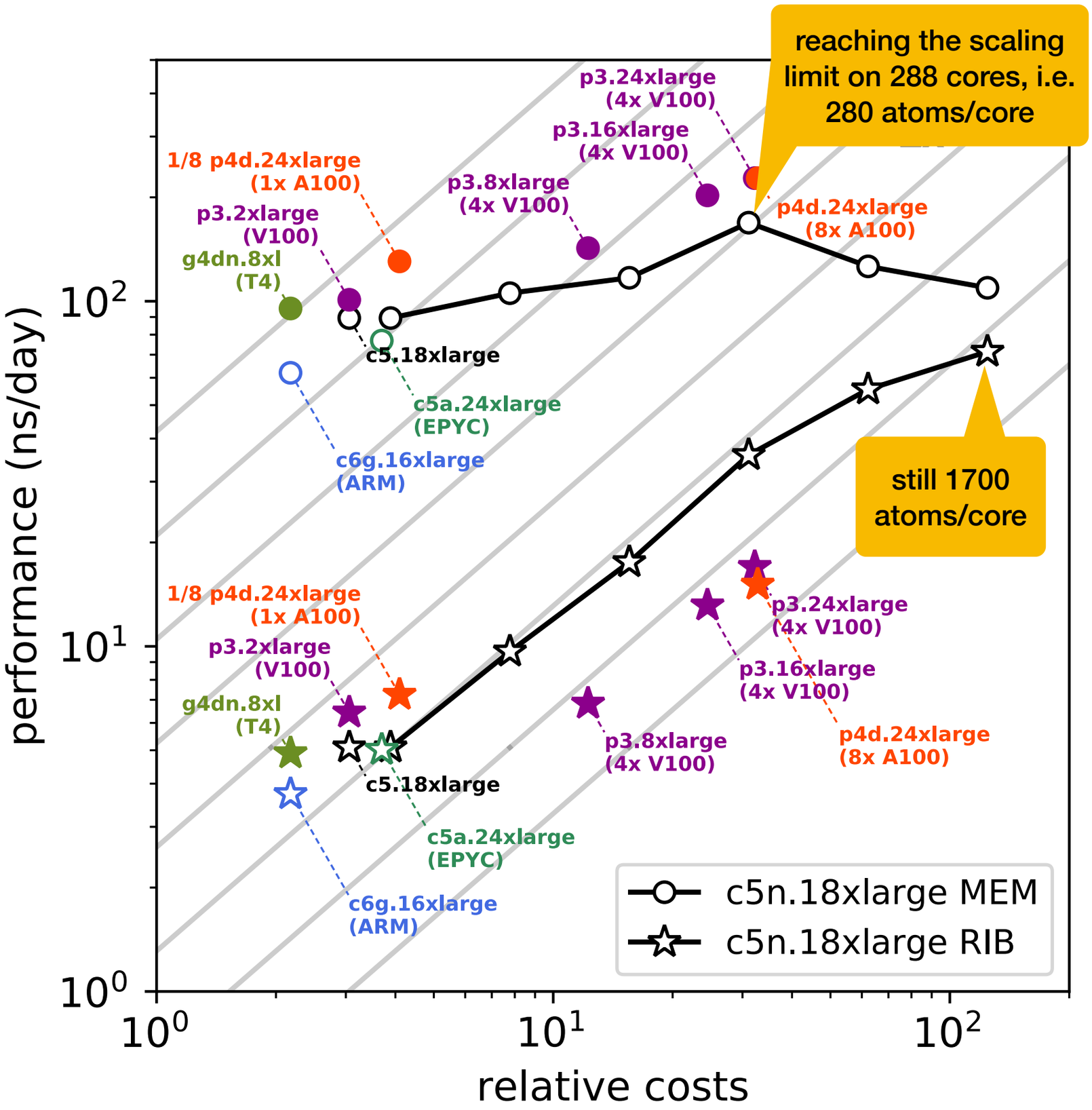
RIB 2 M atoms

GPU

CPU



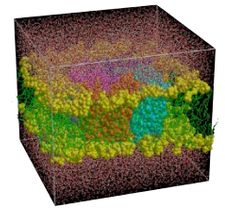
Results a): HPC with GROMACS in the cloud



MEM 81 k atoms

GPU

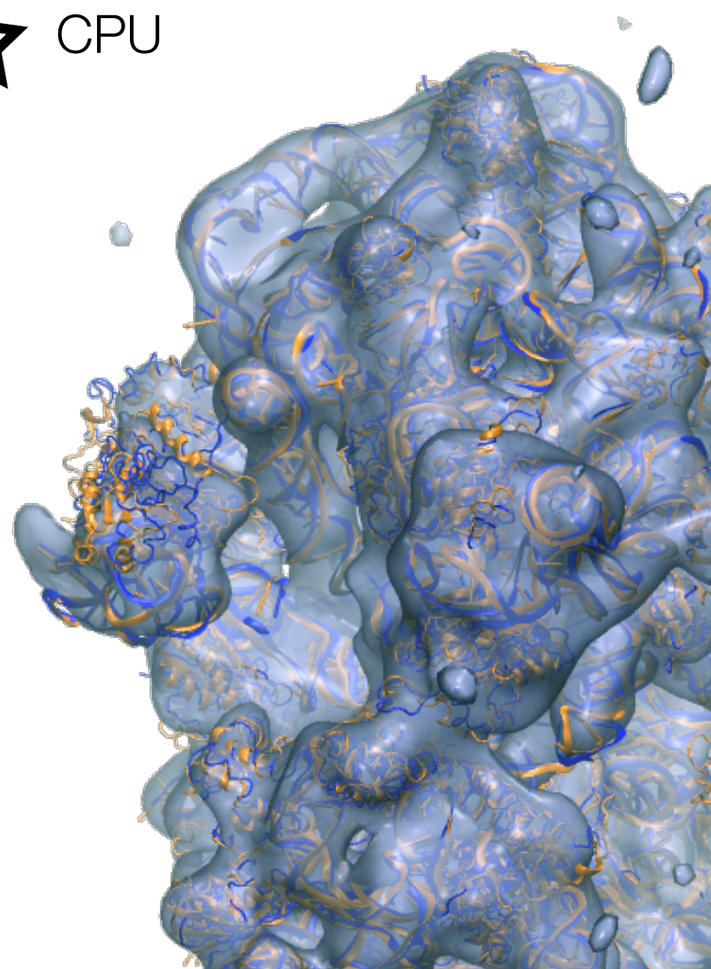
CPU



RIB 2 M atoms

GPU

CPU



WIP b): HTC with GROMACS in the cloud

- now switching to ensemble simulation of 20,000 systems (5,000–110,000 atoms) as used for computational drug design

- this typically takes weeks of runtime on a small cluster

- **Our aim:** speed that up! weeks ➔ overnight (?) in the cloud

    - run ~20,000 jobs (each on one node), wherever there is capacity in the cloud on whatever instance type (using AWS batch + spot instances)

- **The challenges:**

    - Orchestrate runs globally over all regions

    - choose correct software for each available instance (x86, ARM, SIMD capability, GPU yes/no) ➔ containers

    - Different run time of jobs, time-to-solution can not be smaller than longest individual runtime ➔ run large jobs on high-performance instances, small jobs on cheap instances

WIP b): HTC with GROMACS in the cloud

- again, we determined the optimal parameters (ranks x threads) looking at single nodes of all types:
  - Intel and AMD (8–96 cores)
  - ARM Graviton2 (8–64 cores)
  - Nodes with V100 and A100 GPU(s)
- Criteria:
  - performance (for large jobs)
  - performance to price ratio (for smaller jobs)

instance type	pricing ($/h)	ranks × threads	HIF2A performance (ns/d)	(ns/$)
`c6g.16xlarge` 64 vCPUs	2.176	1 × 64	31.870	0.610
	2.176	2 × 32	40.724	0.780
	2.176	4 × 16	47.967	0.918
	2.176	8 × 8	54.431	1.042
	2.176	16 × 4	57.810	1.107
	2.176	32 × 2	58.072	1.112
	2.176	64 × 1	54.645	1.046
`c6g.12xlarge` 48 vCPUs	1.632	1 × 48	30.860	0.788
	1.632	2 × 24	36.544	0.933
	1.632	3 × 16	42.796	1.093
	1.632	4 × 12	43.152	1.102
	1.632	6 × 8	45.375	1.158
	1.632	8 × 6	45.016	1.149
	1.632	12 × 4	46.263	1.181
	1.632	16 × 3	46.132	1.178
	1.632	24 × 2	46.213	1.180
	1.632	48 × 1		
`c6g.8xlarge` 32 vCPUs	1.088	1 × 32	28.694	1.099
	1.088	2 × 16	30.009	1.149
	1.088	4 × 8	32.645	1.250
	1.088	8 × 4	32.448	1.243
	1.088	16 × 2	32.633	1.250
	1.088	32 × 1	33.997	1.302
`c6g.4xlarge` 16 vCPUs	0.544	1 × 16	18.583	1.423
	0.544	2 × 8	17.755	1.360
	0.544	4 × 4	18.361	1.406
	0.544	8 × 2	17.834	1.366
	0.544	16 × 1		
`c6g.2xlarge` 8 vCPUs	0.272	1 × 8	10.043	1.538
	0.272	2 × 4	9.747	1.493
	0.272	4 × 2	9.794	1.500
	0.272	8 × 1	9.701	1.486

WIP b): HTC with GROMACS in the cloud

- again, we determined the optimal parameters (ranks x threads) looking at single nodes of all types:
  - Intel and AMD (8–96 cores)
  - ARM Graviton2 (8–64 cores)
  - Nodes with V100 and A100 GPU(s)
- Criteria:
  - performance (for large jobs)
  - performance to price ratio (for smaller jobs)

instance type	pricing ($/h)	ranks × threads	HIF2A performance (ns/d)	(ns/$)
`c6g.16xlarge`	2.176	1 × 64	31.870	0.610
64 vCPUs	2.176	2 × 32	40.724	0.780
	2.176	4 × 16	47.967	0.918
	2.176	8 × 8	54.431	1.042
	2.176	16 × 4	57.810	1.107
	2.176	32 × 2	58.072	1.112
	2.176	64 × 1	54.645	1.046
`c6g.`		1 × 48	30.860	0.788
48 vCPUs	1.632	2 × 24	36.544	0.933
	1.632	3 × 16	42.796	1.093
	1.632	4 × 12	43.152	1.102
	1.632	6 × 8	45.375	1.158
	1.632	8 × 6	45.016	1.149
	1.632	12 × 4	46.263	1.181
	1.632	16 × 3	46.132	1.178
	1.632	24 × 2	46.213	1.180
	1.632	48 × 1		
`c6g.8xlarge`	1.088	1 × 32	28.694	1.099
32 vCPUs	1.088	2 × 16	30.009	1.149
	1.088	4 × 8	32.645	1.250
	1.088	8 × 4	32.448	1.243
	1.088	16 × 2	32.633	1.250
	1.088	32 × 1	33.997	1.302
`c6g.4xlarge`	0.544	1 × 16	18.583	1.423
16 vCPUs	0.5		17.755	1.360
	0.5		18.361	1.406
	0.5		17.834	1.366
	0.5			
`c6g.2xlarge`	0.2		043	1.538
8 vCPUs	0.272	2 × 4	9.747	1.493
	0.272	4 × 2	9.794	1.500
	0.272	8 × 1	9.701	1.486

hightest performance

best performance to price ratio

Summary

- Setting up a cloud-based HPC cluster to run scientific simulations on is straightforward – demonstrated with GROMACS

- Cluster installation (AWS ParallelCluster) is separate from software installation (Spack), both workflows are easily reproducible

- Advantages of cloud-based computing:
    - Diverse hardware readily available. Various architectures (Intel, AMD, ARM) in various sizes (#of cores) & combinations (network, accellerators)
    - Hardware variability allows to optimize for time-to-solution or performance-to-price
    - No waiting time for jobs in the queue due to dynamic scaling
- GROMACS = our showcase. Approach should work for diverse scientific software

Acknowledgments

Department of Theoretical & Computational Biophysics
@ MPI for Biophysical Chemistry Göttingen

The AWS Team

Christian Kniep, Thorsten Bloth, Stephen Sachs, Johannes Schulz