Contact:
ckutzne@mpinat.mpg.de

GROMACS in the Cloud

A global supercomputer to speed up alchemical drug design

C Kutzner,[1] C Kniep,[2] A Cherian,[2] L Nordstrom,[2] H Grubmüller,[1] BL de Groot,[1] and V Gapsys[1]

[1]MPI for Multidisciplinary Sciences, Theoretical & Computational Biophysics, Göttingen [2]Amazon Web Services, Berlin, Singapore, London

MAX PLANCK INSTITUTE
FOR MULTIDISCIPLINARY SCIENCES

Abstract

- **Cloud-based** computing as offered by, e.g., Amazon Web Services (AWS), Microsoft Azure, and Google Cloud is increasingly adopted for **scientific applications**[1,2]
- The hardware on which scientific simulations codes run should (a) enable a high simulation **performance**, (b) allow for good parallel **scaling**, (c) have a high **cost-efficiency**
- We assess **costs and efficiency of state-of-the-art cloud computing** and compare the results to a traditional on-premises compute cluster. Our use case are **biomolecular simulations** carried out with the **GROMACS**[3] molecular dynamics (MD) toolkit
- We set up an HPC cluster in the AWS cloud with various instances (Intel, AMD, and ARM CPUs, NVIDIA GPUs)
- We benchmark how GROMACS performs on individual instances and how it scales across multiple instances. We assess which instances are most cost-efficient for producing MD trajectories
- We find that, in terms of total costs including hardware, personnel, room, energy and cooling, MD in the **cloud is as cost-efficient as an on-premises cluster**[4] given that optimal cloud instances are chosen
- We demonstrate that high-throughput ligand-screening, as used in **computational drug design**, can be **massively accelerated by using global cloud resources**. For a ligand screening study[5,6] consisting of ~20,000 independent simulations or 200 μs of combined simulation trajectory, we made use of diverse hardware available in the cloud at the time of the study. Using more than 4,000 instances, 140,000 cores, and 3,000 GPUs around the globe, our simulation ensemble that would normally take weeks to complete on a typical on-premises cluster consisting of several hundred nodes, finished in about two days in the cloud

Methods

Setting up a cloudy cluster

- We use **ParallelCluster** (https://github.com/aws/aws-parallelcluster) as open source, free, cluster management tool
- We use **Spack** as flexible package manager for HPC software (https://github.com/spack/spack.git)
- With GPU support via CUDA 10.2 and IntelMPI 2019:
  `spack install gromacs@2020.2 +cuda ^intel-mpi`
- GROMACS 2020 (for CPU runs) and 2021 (for GPU runs and on hpc6a)
- Benchmark method as described in Ref. 4

Accessing global resources

- **Aim:** Minimize time-to-solution for a drug design ensemble consisting of 19,872 simulations[5,6] (5,000–100,000 atoms)
- **Approach:** Run all simulations at the same time, each on a separate Spot instance, wherever there is capacity in the cloud
- Used **AWS Cyclone** to provide a single entry point for jobs scheuled to AWS Batch queues around the globe (Fig. 4)
- Cyclone is an open source community-supported solution https://github.com/awslabs/aws-cyclone-solution
- Cyclone provides cross-regional serverless HPC job scheduling and resource orchestration using DynamoDB, Lambda functions, Step functions, AWS Kinesis Data Streams, the Simple Queue Service (SQS), and the Amazon API Gateway
- State of jobs is listed in a DynamoDB table, while a Kibana dashboard shows the total number of running instances, the instance types by region, and more

Conclusions

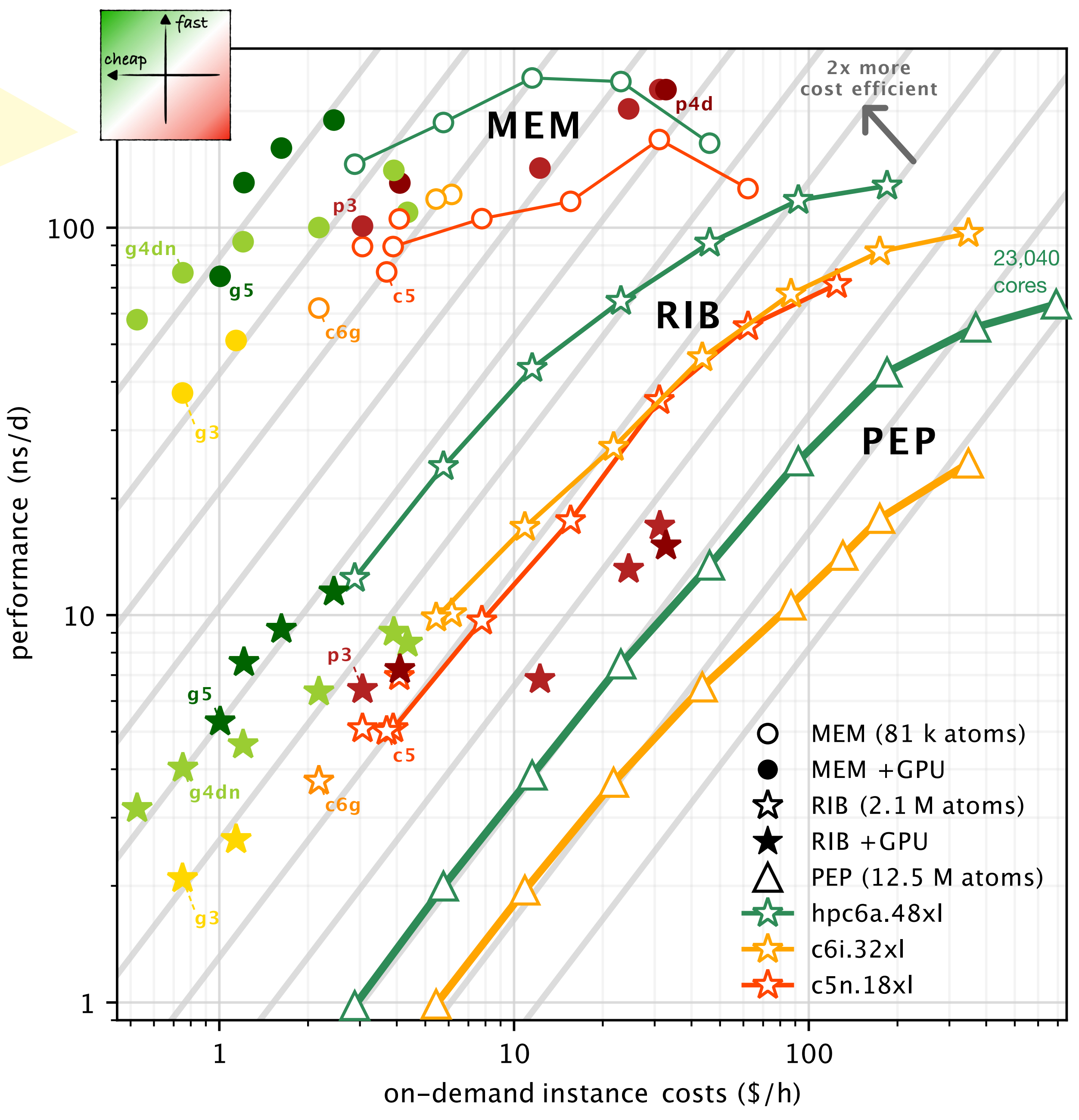
- The cloud offers a **wide range of instance types** (2–128 vCPU Intel, AMD & ARM instances, optionally with GPUs and fast interconnect, so that for each application optimal hardware can be picked
- **Cloud-based HPC is feasible:** large MD systems scale well on instances with fast interconnect, as demonstrated by strong scaling on up to 23,040 hpc6a cores (Fig. 1)
- Cost-efficiency depends critically on choosing appropriate instance types: For GROMACS on AWS, **g5** and **g4dn** provide optimal cost-efficiency on individual instances, followed by **hpc6a** for scaling scenarios (Fig. 1)
- The costs of cloud computing and on-premises computing have become similar (Figs. 2–4). Compared to spot instances with optimal cost-efficiency (g5.2xl), **an on-premises cluster aggressively tuned for GROMACS can produce 1.2× as much trajectory per €** (Fig. 4)
- Our AWS Cyclone-based workflow **distributes large-scale simulation projects globally**, thus dramatically decreasing the time-to-solution (Fig. 5)
- Our **computational drug design** ensemble[5,6] consisting of 20,000 simulations acquired 140,000 compute cores and 3,000 GPUs on 4,000 instances shortly after submitting. The whole ensemble finished in 2 days, as opposed to requiring weeks on a typical on-premises cluster consisting of several hundred nodes shared among many users (Fig. 6)

References

1. G Guidi, M Ellis, A Buluç, K Yelick, D Culler: *10 years later: Cloud computing is closing the performance gap.* In Companion of the ACM/SPEC Int. Conference on Performance Engineering 41–48 (2021)
2. J Lofstead, D Duplyakin. *Take Me to the Clouds Above: Bridging On Site HPC with Clouds for Capacity Workloads.* CLOUD COMPUTING 2021 (2021): 63
3. S Páll, MJ Abraham, C Kutzner, B Hess, E Lindahl: *Tackling exascale software challenges in molecular dynamics simulations with GROMACS.* EASC 2014, Stockholm, 3–27 (Eds. Markidis, S.; Laure, E.) Springer, Cham (2015)
4. C Kutzner, S Páll, M Fechner, A Esztermann, BL de Groot, H Grubmüller: *More Bang for Your Buck: Improved use of GPU Nodes for GROMACS 2018.* J. Comp. Chem. 40, 27 (2019): 2418–2431
5. V Gapsys, DF Hahn, G Tresadern, DL Mobley, M Rampp, BL de Groot: *Pre-Exascale Computing of Protein-Ligand Binding Free Energies with Open Source Software for Drug Design.* J. Chem. Inf. Model. 62 (5), 1172–1177 (2022)
6. C Kutzner, C Kniep, A Cherian, L Nordstrom, H Grubmüller, BL de Groot, V Gapsys: *GROMACS in the Cloud: A Global Supercomputer to Speed Up Alchemical Drug Design,* J. Chem. Inf. Model. 62 (7), 1691–1711 (2022)

Results I: Choosing optimal instances for MD simulation

Figure 1. GROMACS performance as a function of instance costs for the MEM (circles), RIB (stars) and PEP (triangles) systems on different generations (color) of CPU (open symbols) and GPU instances (filled symbols). Connected symbols indicate strong scaling. The tilted grey lines are **isolines of equal performance to price ratio** with better configurations to the upper left. g5 GPU instances are most cost-efficient for GROMACS and were therfore used for the cost comparison (Figs. 2–4)



Biomolecular benchmark systems:[4]

https://www.mpinat.mpg.de/grubmueller/bench

MEM 81 k atoms
Aquaporin tetramer in lipid membrane + ions + water, PME electrostatics, 2 fs time step

RIB 2.1 M atoms
Ribosome in solution, PME, 4 fs time step

PEP 12.5 M atoms
Peptides in water, PME, 2 fs time step

Acknowledgments

Compute time for all cloud-based simulations of this study was generously provided by AWS public sector. Many thanks to Torsten Bloth, Stephen Sachs, Cristian Magherusan-Stanciu, Brendan Bouffler, Bruno Silva, Agata Jablonka, and Johannes Schulz for advice and support throughout the project. The total costs of rack space were estimated by A. Esztermann. Simulation input preparation and output analysis was done on compute clusters of the Max Planck Society. The work was supported by the BioExcel CoE (www.bioexcel.eu), a project funded by the European Union contracts H2020-INFRAEDI-02-2018-823830.

Results II: Isn't a local cluster cheaper?

Figure 2. Comparing cloud costs to on-premises compute nodes over 3 years. Violet bars BC show cloud instances with best cost-efficiency for GROMACS (see Fig. 1), compared to on-premises nodes AD.

A Consumer GPU node[4] optimized for GROMACS usage.
B Spot g5.2xl GPU instance.
C Reserved g5.2xl GPU instance (3 years, upfront payment).
D As A, but for a 4 U node with professional GPU (e.g. Quadro P6000).

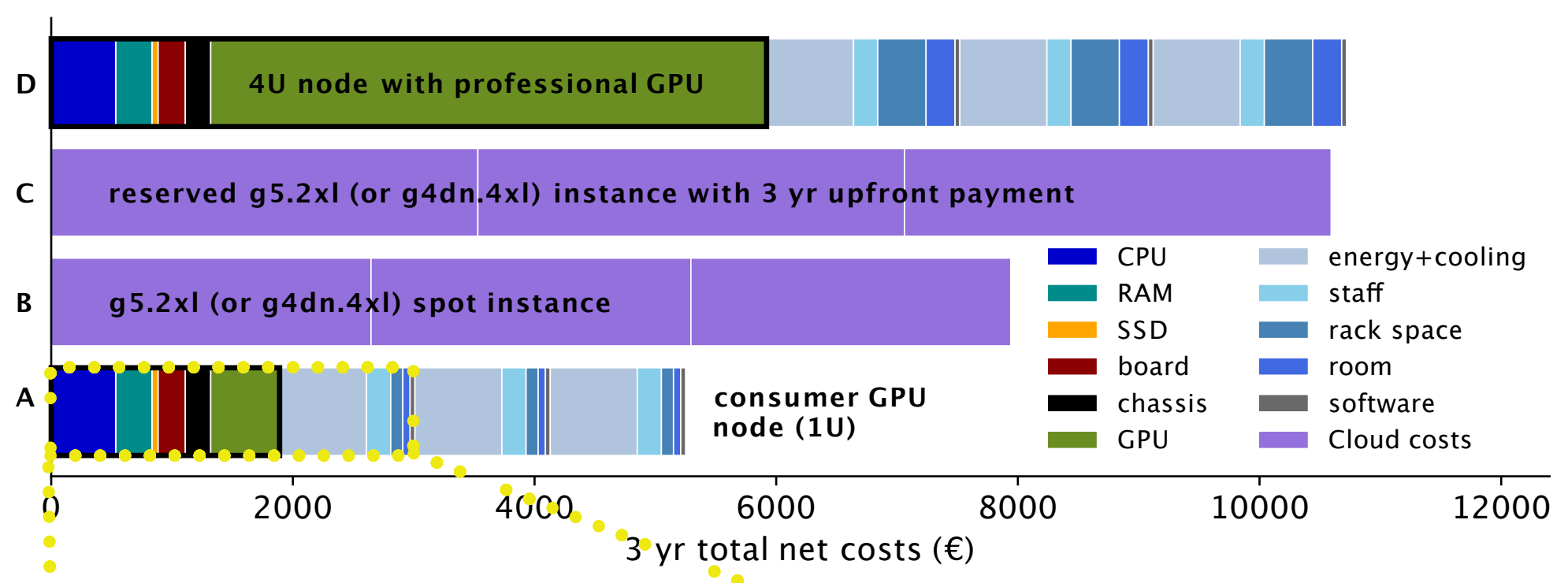


Figure 3. Breakdown of total node costs for GROMACS-optimized on-premises cluster (bar A in Fig. 2) for the first year of operation.

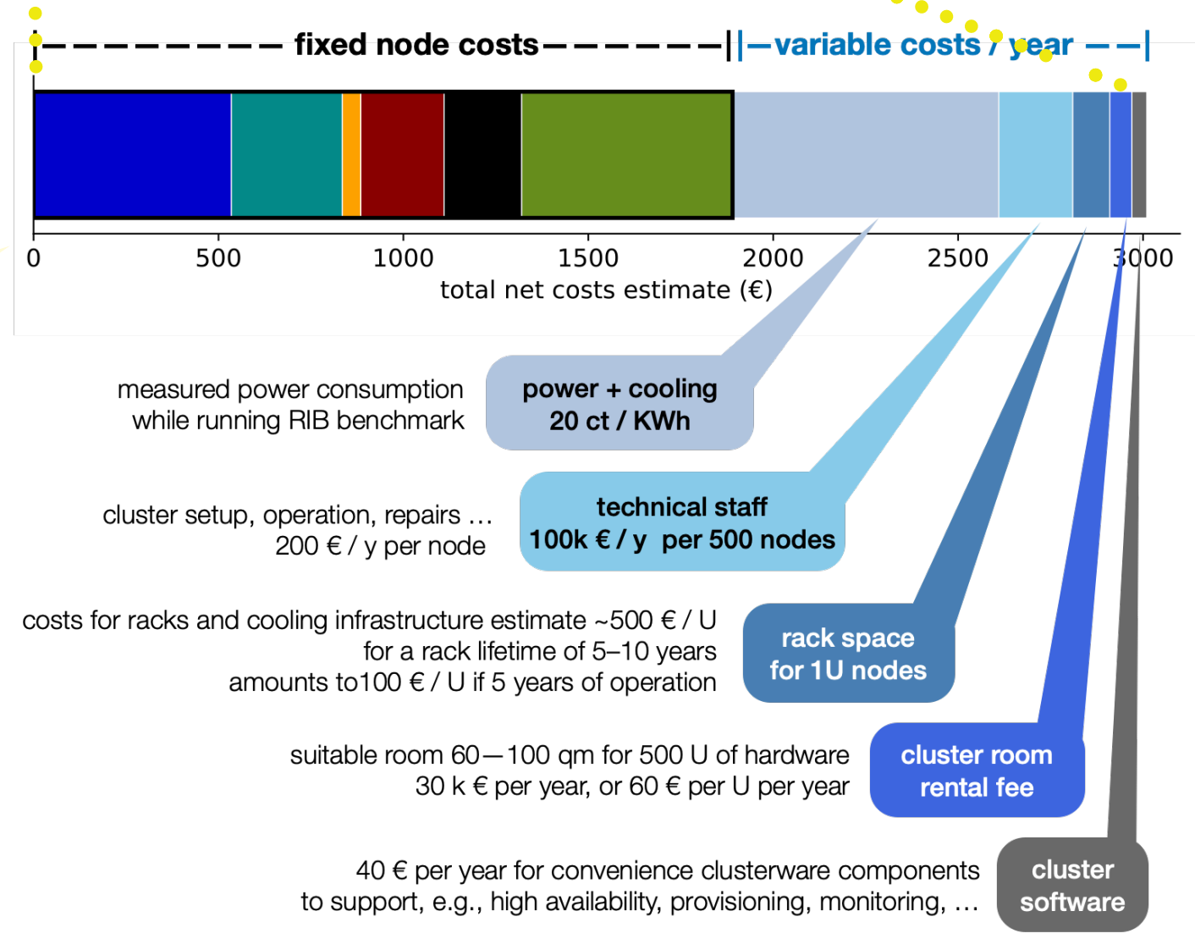
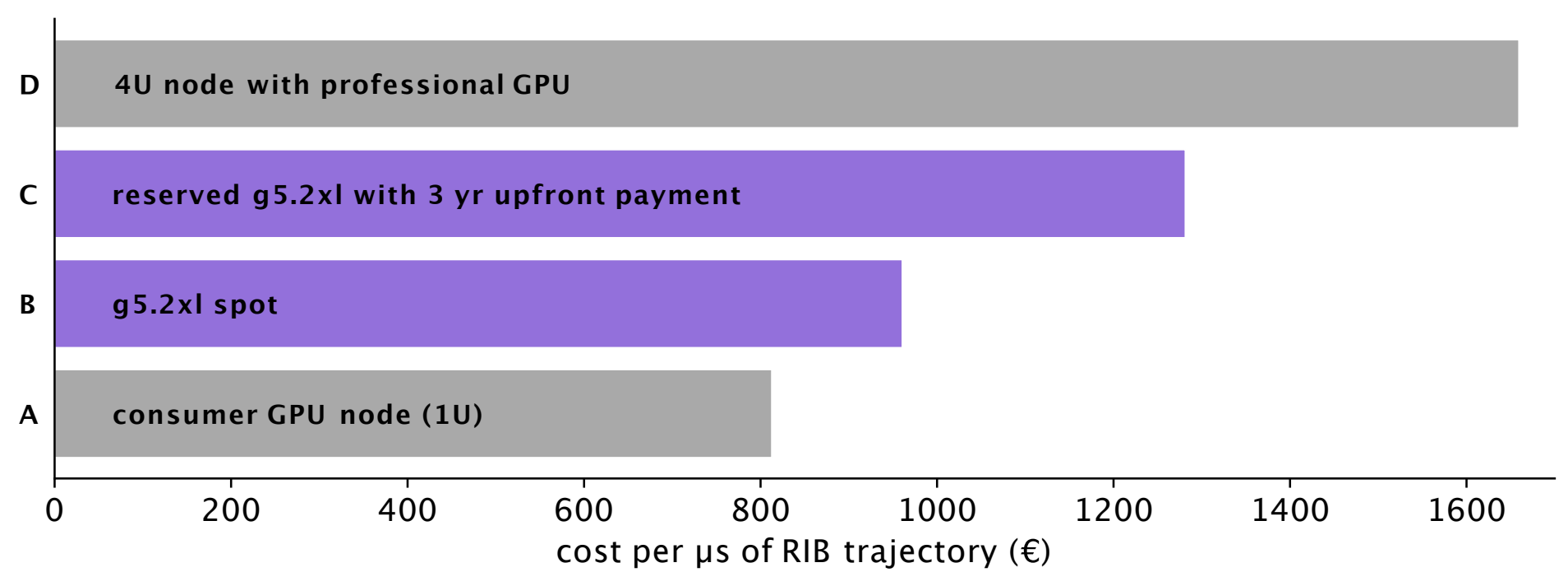


Figure 4. Cost-efficiency of cloud instances vs. on-premises nodes for 3 years of operation. Same configurations as in Fig. 2, but now comparing the total costs per μs of RIB trajectory.



Cost comparison

- We consider all costs that arise for operating an exemplary 500 node on-premises cluster over 3 years
- Compute costs only, we are not considering costs for storage of trajectories
- Net costs to produce 1 μs of RIB trajectory
- We tune our on-premises cluster aggressively for throughput with GROMACS, by **using GeForce consumer GPUs**, reasonably priced CPUs, omitting HPC interconnects, a minimal amount of RAM, and dense packing (~1 GPU per U)
- In an earlier study[4] we demonstrated that this leads to trajectory costs of 1/3× the trajectory costs of CPU nodes or nodes with Tesla GPUs
- Our **optimized 1U consumer GPU node** (Fig. 3A, 20 hardware threads + RTX 2080 GPU) has total net costs of 5250 € for 3 years of operation and produces 5.9 ns/d of RIB trajectory, i.e. 6.46 μs over 3 years. This leads to **trajectory costs of ~810 €/μs** (bar A, lower panel).
- **AWS g5.2xl instances** (Fig. 3B-C, 8 vCPUs + A10G GPU) offer a RIB performance of 7.55 ns/d at optimal cost-efficiency (Fig. 1 lower left). Whereas **on-demand pricing yields costs of ~3,200 €/μs** (not shown), with **3 year upfront payment that reduces to ~1,300 €/μs** (bar C).
- **g5.2xl spot instances** are available at ~30% of the on-demand price. This yields **costs of ~950 €/μs** (bar B), approaching those of our on-premises cluster tailored towards GROMACS.

Results III: Global compute resources accelerate drug design

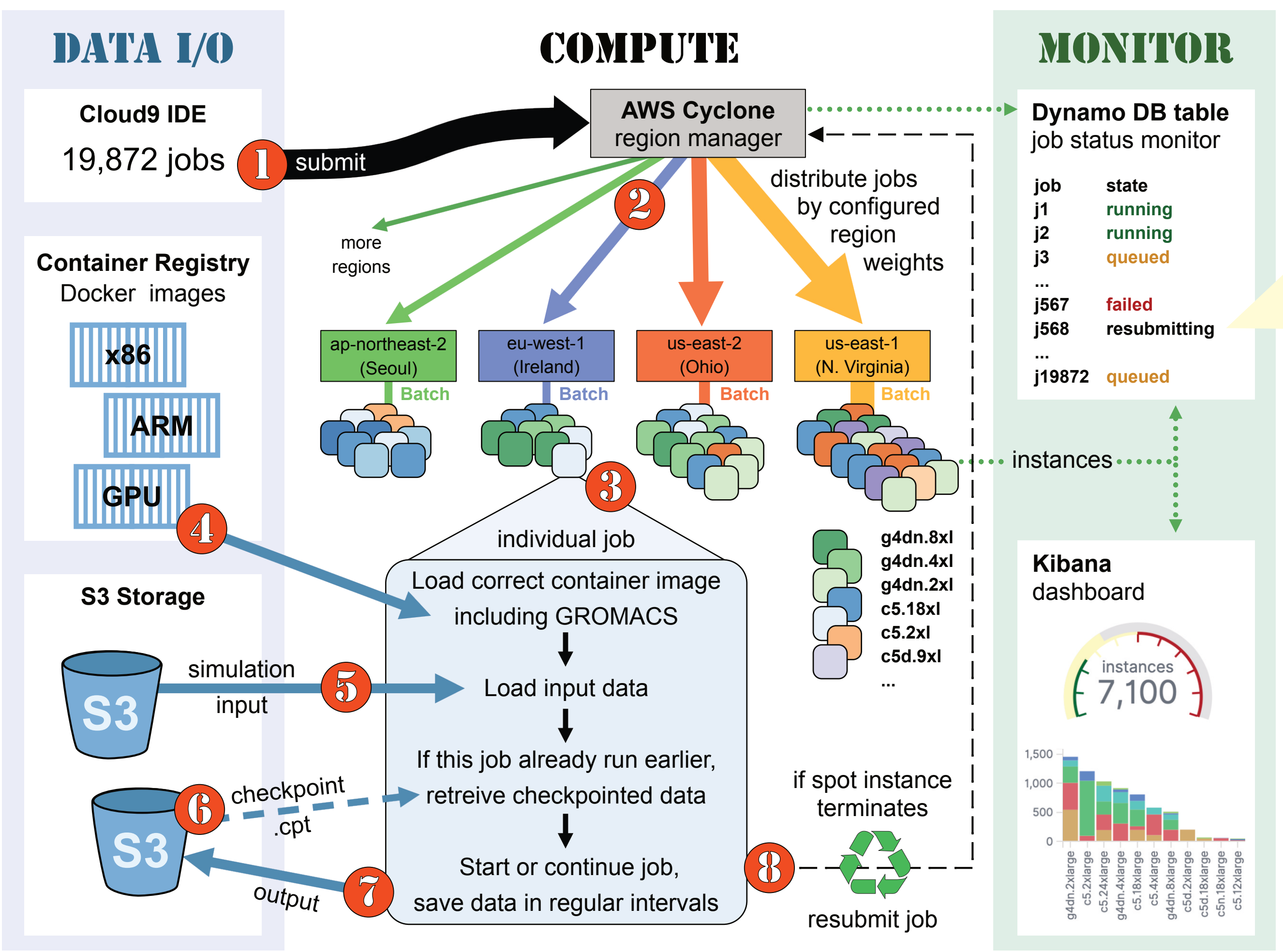
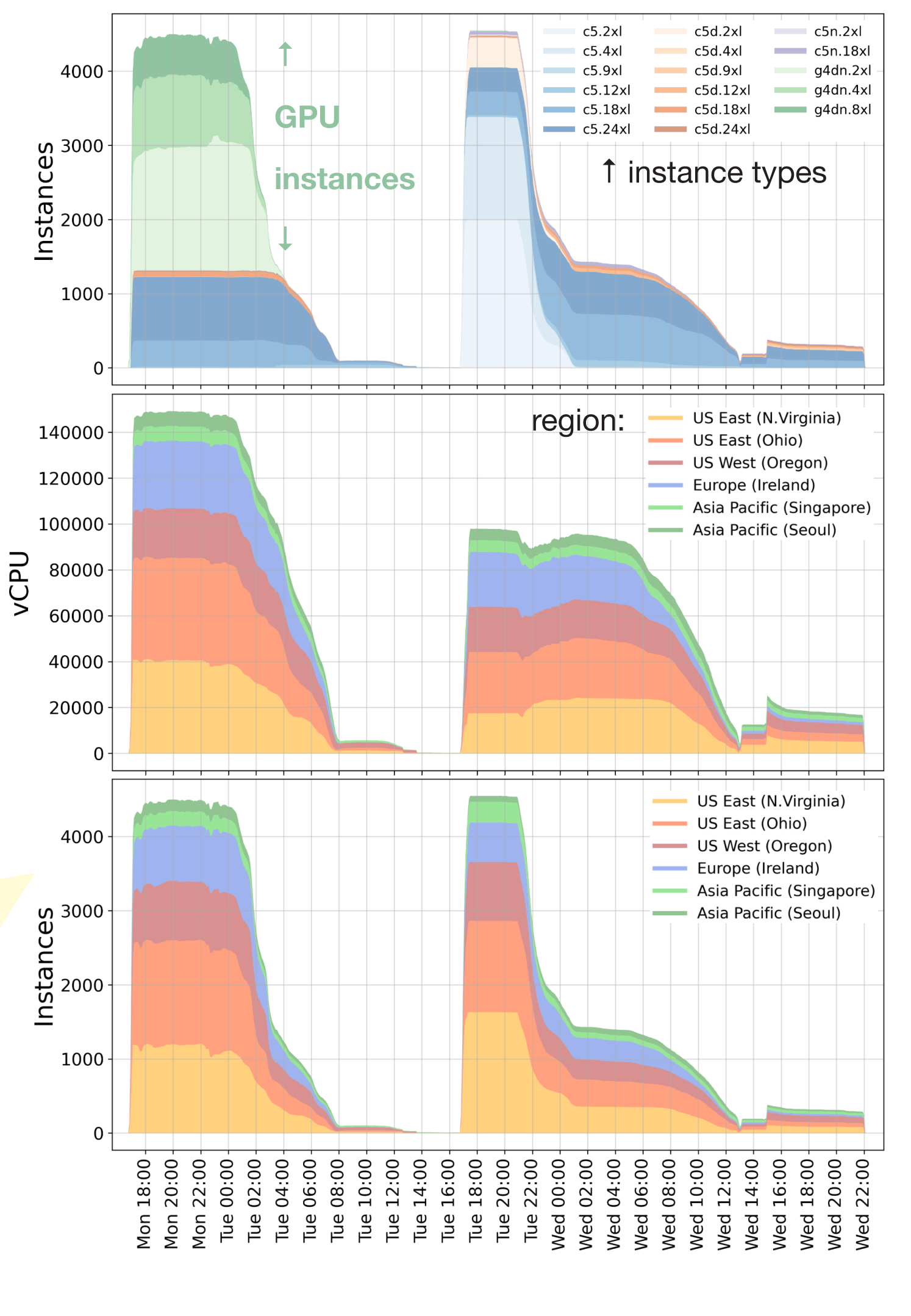


Figure 5. Jobs are distributed to globally available compute resources using AWS Cyclone.

19,872 jobs are submitted to Cyclone (1), to be distributed globally (2). An individual job is started (3) on a Batch Spot instance with sufficient resources (vCPUs, GPUs). The instance loads the correct Docker image from AWS ECR (4) with preinstalled software. Simulation input files are loaded fom S3 (5), possibly also checkpoint data (6). Output is written to S3 (7), including intermediate checkpoints. Spot instance termination will trigger job resubmssion (8).

Figure 6. Usage of global compute resources over time for the ligand screening study.
Top: Colors show the various instances that were acquired globally over time.
Middle: Total vCPUs by regions.
Bottom: Number of instances by region.


Instances
GPU instances
↑ instance types
c5.2xl
c5.4xl
c5.9xl
c5.12xl
c5.18xl
c5.24xl
c5d.2xl
c5d.4xl
c5d.9xl
c5d.12xl
c5d.18xl
c5d.24xl
c5n.2xl
c5n.18xl
g4dn.2xl
g4dn.4xl
g4dn.8xl
vCPU
region:
US East (N.Virginia)
US East (Ohio)
US West (Oregon)
Europe (Ireland)
Asia Pacific (Singapore)
Asia Pacific (Seoul)
Instances
Mon 18:00
Mon 20:00
Mon 22:00
Tue 00:00
Tue 02:00
Tue 04:00
Tue 06:00
Tue 08:00
Tue 10:00
Tue 12:00
Tue 14:00
Tue 16:00
Tue 18:00
Tue 20:00
Tue 22:00
Wed 00:00
Wed 02:00
Wed 04:00
Wed 06:00
Wed 08:00
Wed 10:00
Wed 12:00
Wed 14:00
Wed 16:00
Wed 18:00
Wed 20:00
Wed 22:00