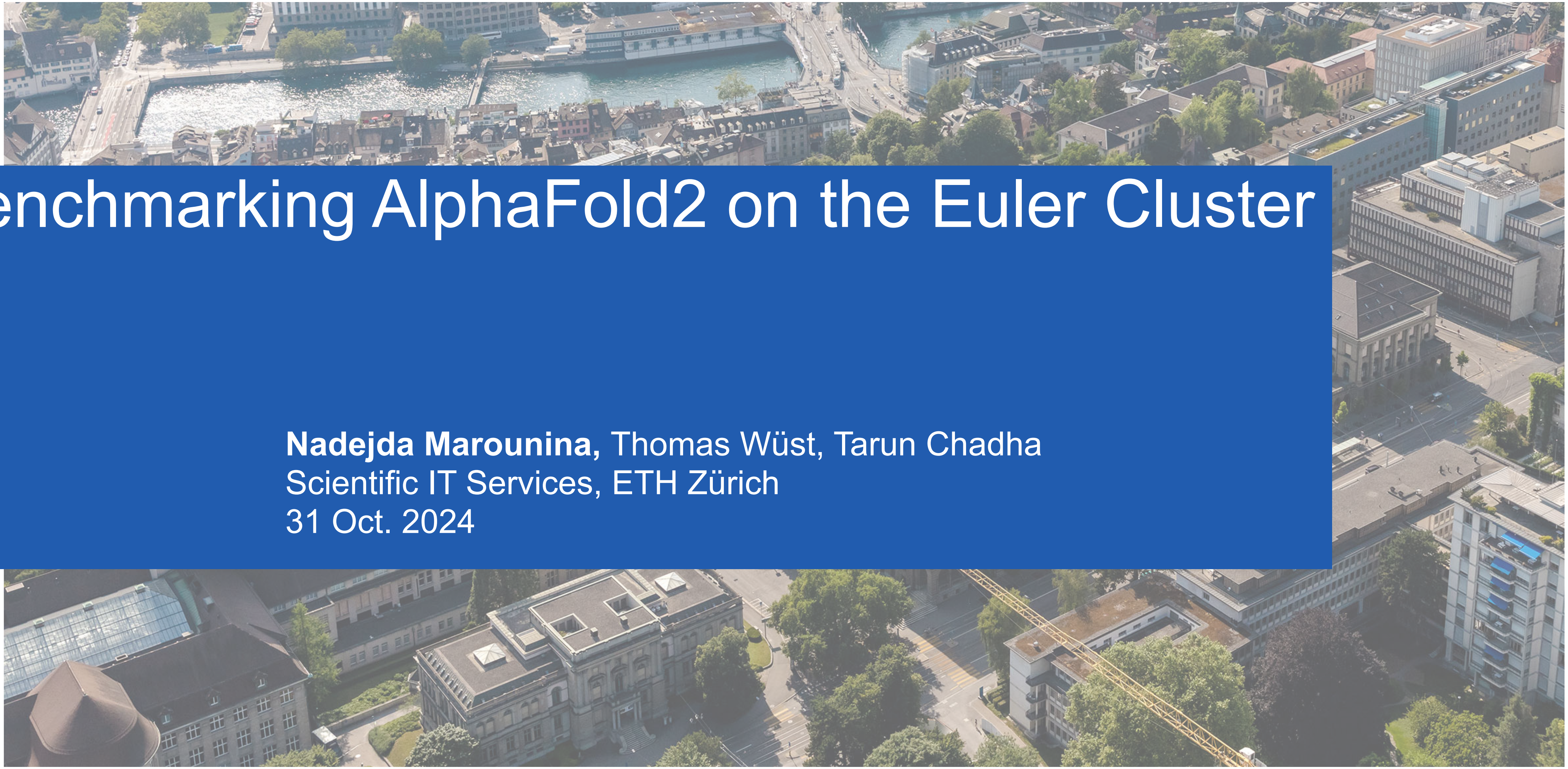
ETH zürich



Benchmarking AlphaFold2 on the Euler Cluster

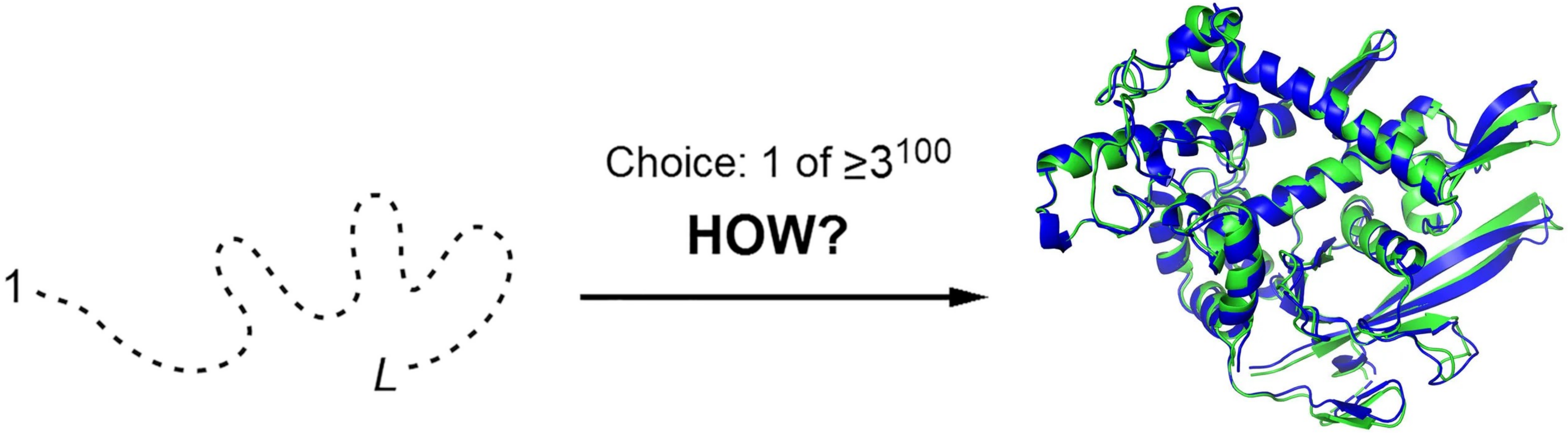
Nadejda Marounina, Thomas Wüst, Tarun Chadha
Scientific IT Services, ETH Zürich
31 Oct. 2024

Overview:

- What is AlphaFold2 and why is it important ?
- How does AlphaFold2 work ?
- How AlphaFold2 uses HPC resources ?

What is AlphaFold2 ?

The protein folding problem



Choice: 1 of ≥3^{100}

HOW?

Unfolded protein chain: ≥3^{100} conformations
for a chain of L=100 amino acid residues

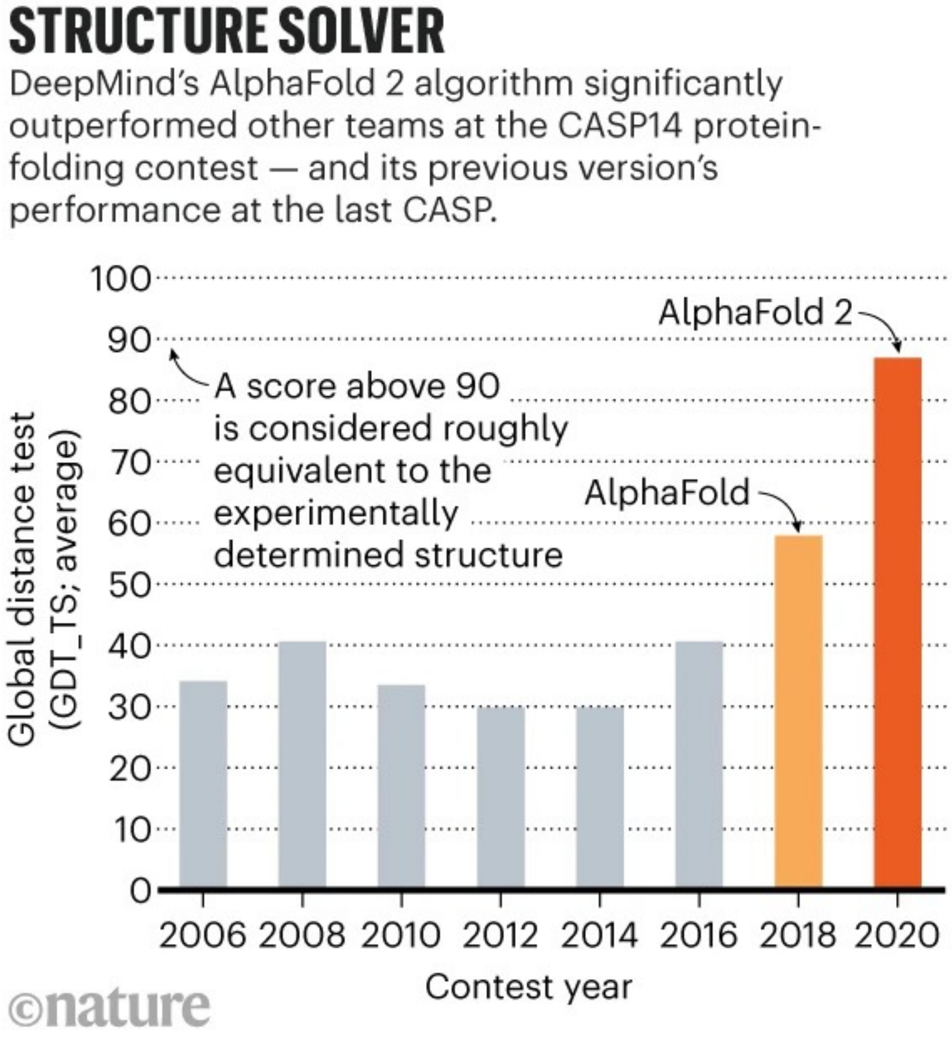
Native protein structure:
1 conformation

- Each protein fulfils a **specific (and usually vital) function** thanks to its specific **shape**
- Experimental measurements are time-consuming and complicated (as of 2024 : >200k 3D structures experimentally confirmed; ~230M amino acid sequences discovered so far)

Image from : Finkelstein, A.V., et al. Protein folding problem: enigma, paradox, solution. Biophys Rev 14, 1255–1272 (2022), modified

CASP : Critical Assessment of Structure Prediction

- CASP aims to establish the current state of the art in protein structure prediction
- Pre-2018, models were generally based on physical principles (template-based modelling or free modelling)
- **AlphaFold2** proposes a **machine learning** approach : it is able to predict a protein shape, but not to explain how/why the protein gets this shape



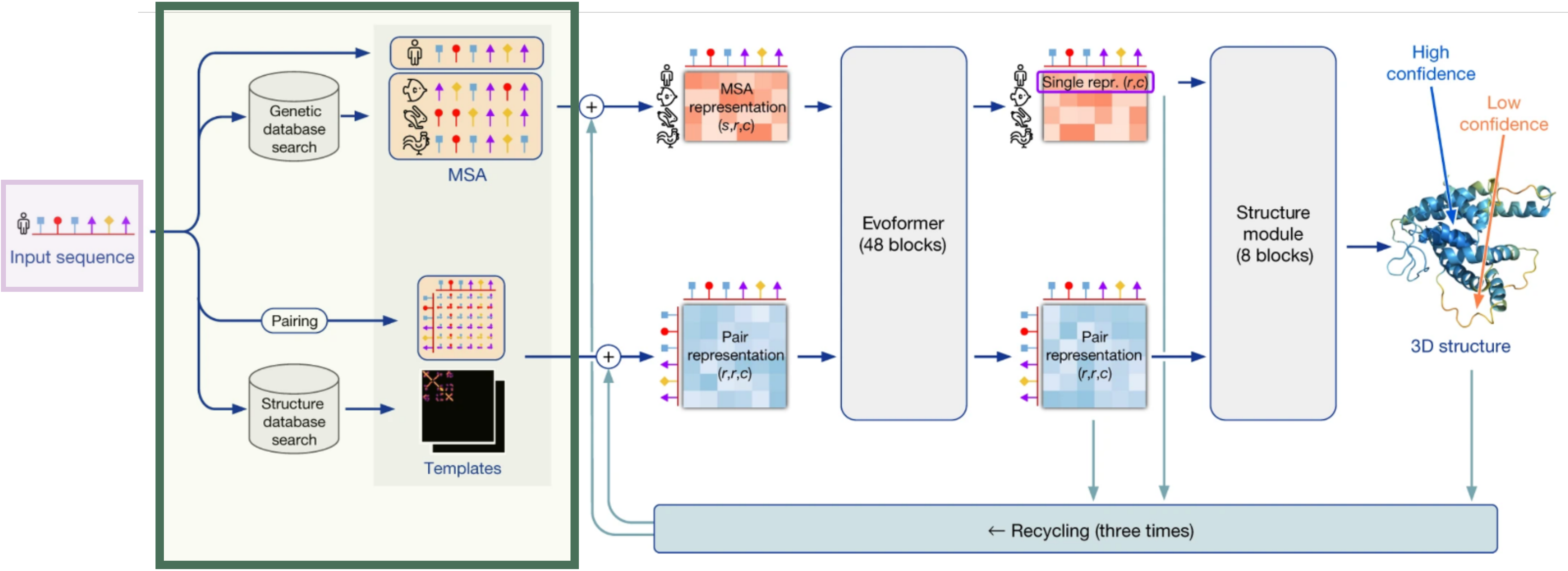
Bertoline LMF, Lima AN, Krieger JE, Teixeira SK. Before and after AlphaFold2: An overview of protein structure prediction. Front Bioinform. 2023 Feb 28;3:1120370.

Why talking about AlphaFold2 here ?

- **AlphaFold3** has been currently released as a **closed source software** on a server, with free access up to 5000 tokens and 20 jobs per 24h
- While AlphaFold3 claims to have the best accuracy, the previous version of the software stays relevant as the accuracy of AF2 is still high
- However use cases of the software on the Euler cluster include folding a few thousand of proteins (up to 100k), or very long proteins that can exceed 5000 amino acids
- Therefore, **AlphaFold2 stays a relatively popular software**, used on a daily basis on the cluster
- It’s a ML application that prompted a lot of work for its installation and operation on the cluster and also created a lot of exchanges with our users to make sure they avoid some HPC pitfalls and use the HPC resource in a smart way

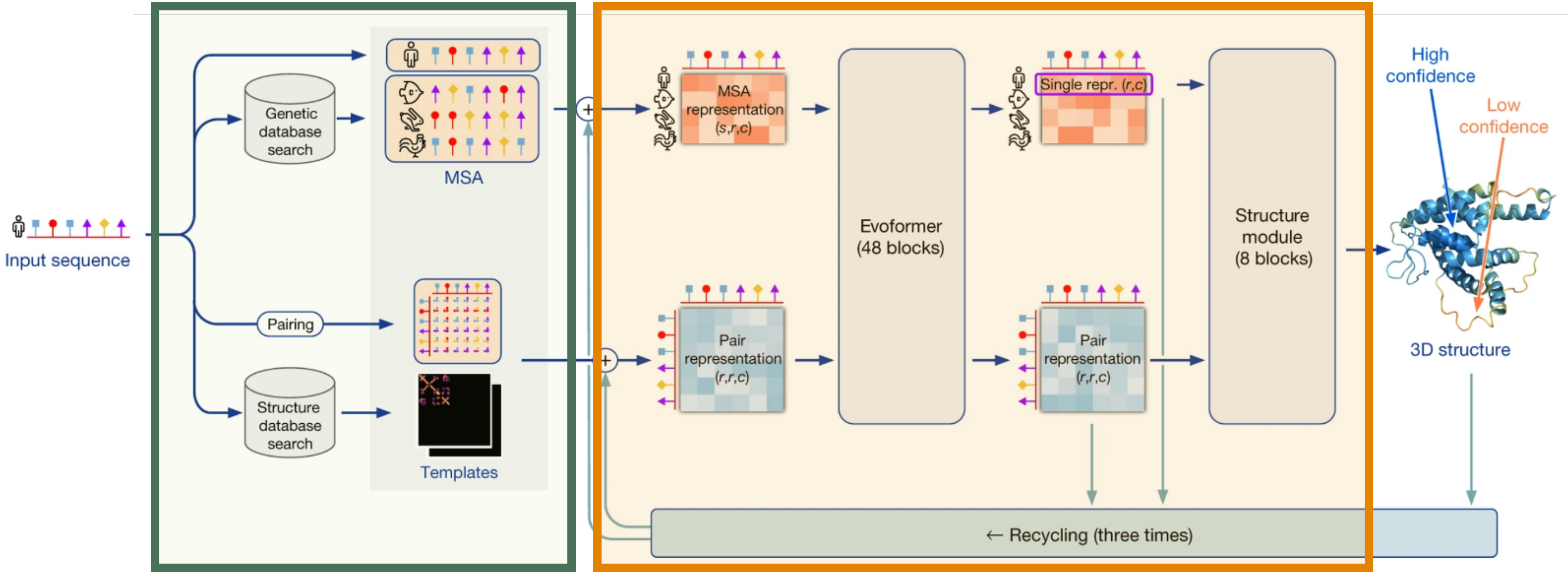
How does AlphaFold2 works ?
(Very brief)

“Big picture” of the algorithm structure :


Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Data gathering

“Big picture” of the algorithm structure :



Input sequence
Genetic database search
MSA
Pairing
Structure database search
Templates
MSA representation (s,r,c)
Pair representation (r,r,c)
Evoformer (48 blocks)
Single repr. (r,c)
Pair representation (r,r,c)
Structure module (8 blocks)
High confidence
Low confidence
3D structure
← Recycling (three times)

Data gathering (CPU)

Inference on the structure (GPU)

GPU models on Euler :

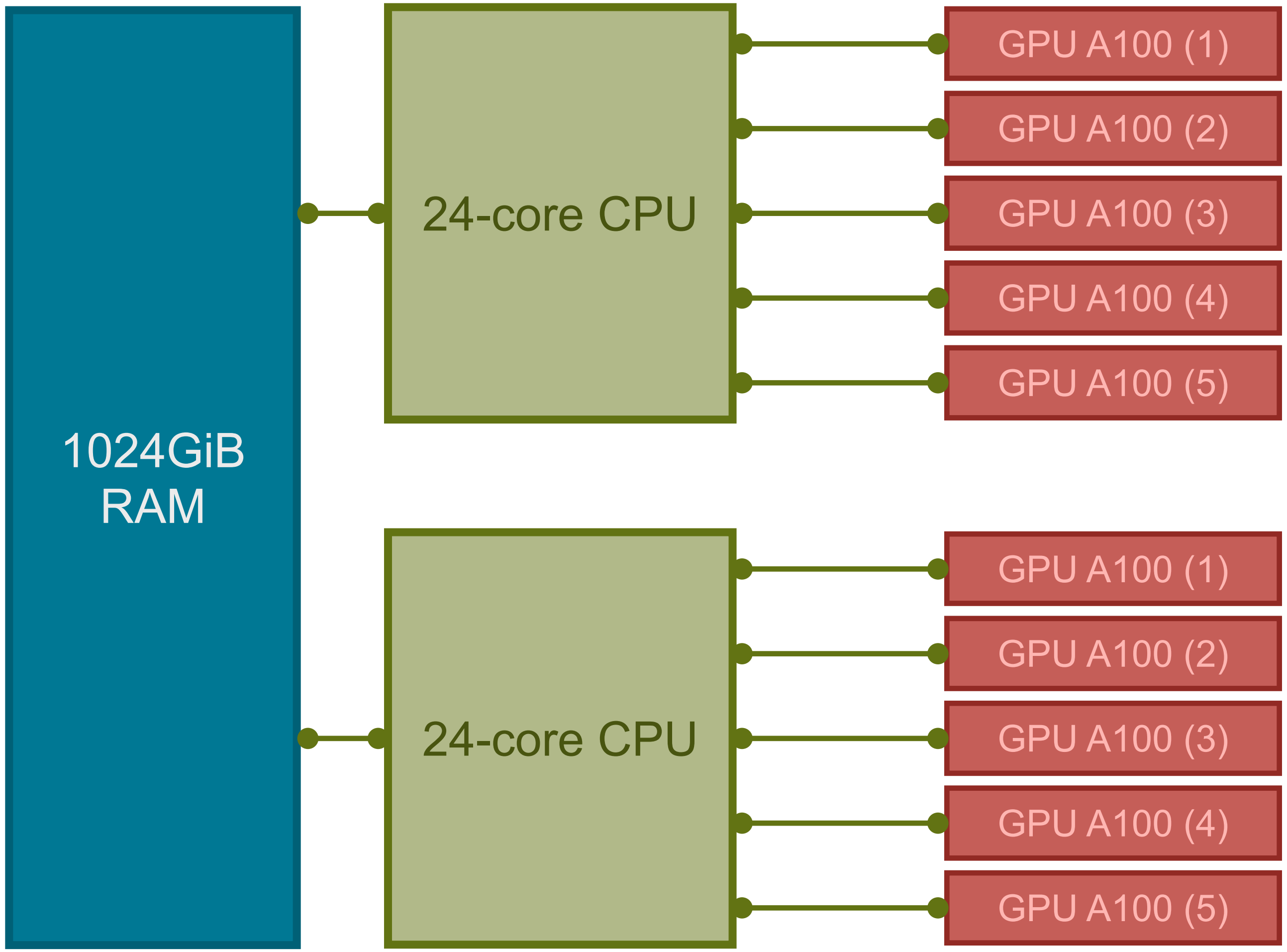
Shorter pending times

Longer pending times

GPU model	SLURM specifier	GPU memory	CPU cores	CPU memory
NVIDIA GeForce GTX 1080 Ti	gtx_1080_ti	11 GiB	20	256 GiB
NVIDIA GeForce RTX 2080 Ti	rtx_2080_ti	11 GiB	36	384 GiB
NVIDIA GeForce RTX 2080 Ti	rtx_2080_ti	11 GiB	128	512 GiB
NVIDIA GeForce RTX 3090	rtx_3090	24 GiB	128	512 GiB
NVIDIA GeForce RTX 4090	rtx_4090	24 GiB	128	512 GiB
NVIDIA TITAN RTX	titan_ttx	24 GiB	128	512 GiB
NVIDIA Quadro RTX 6000	quadro_rtx_6000	24 GiB	128	512 GiB
NVIDIA Tesla V100-SXM2 32 GiB	v100	32 GiB	48	768 GiB
NVIDIA Tesla V100-SXM2 32 GB	v100	32 GiB	40	512 GiB
Nvidia Tesla A100 (40 GiB)	a100-pcie-40gb	40 GiB	48	768 GiB
Nvidia Tesla A100 (80 GiB)	a100_80gb	80 GiB	48	1024 GiB

https://scicomp.ethz.ch/wiki/GPU_job_submission_with_SLURM

A very common pitfall in resource requirement :

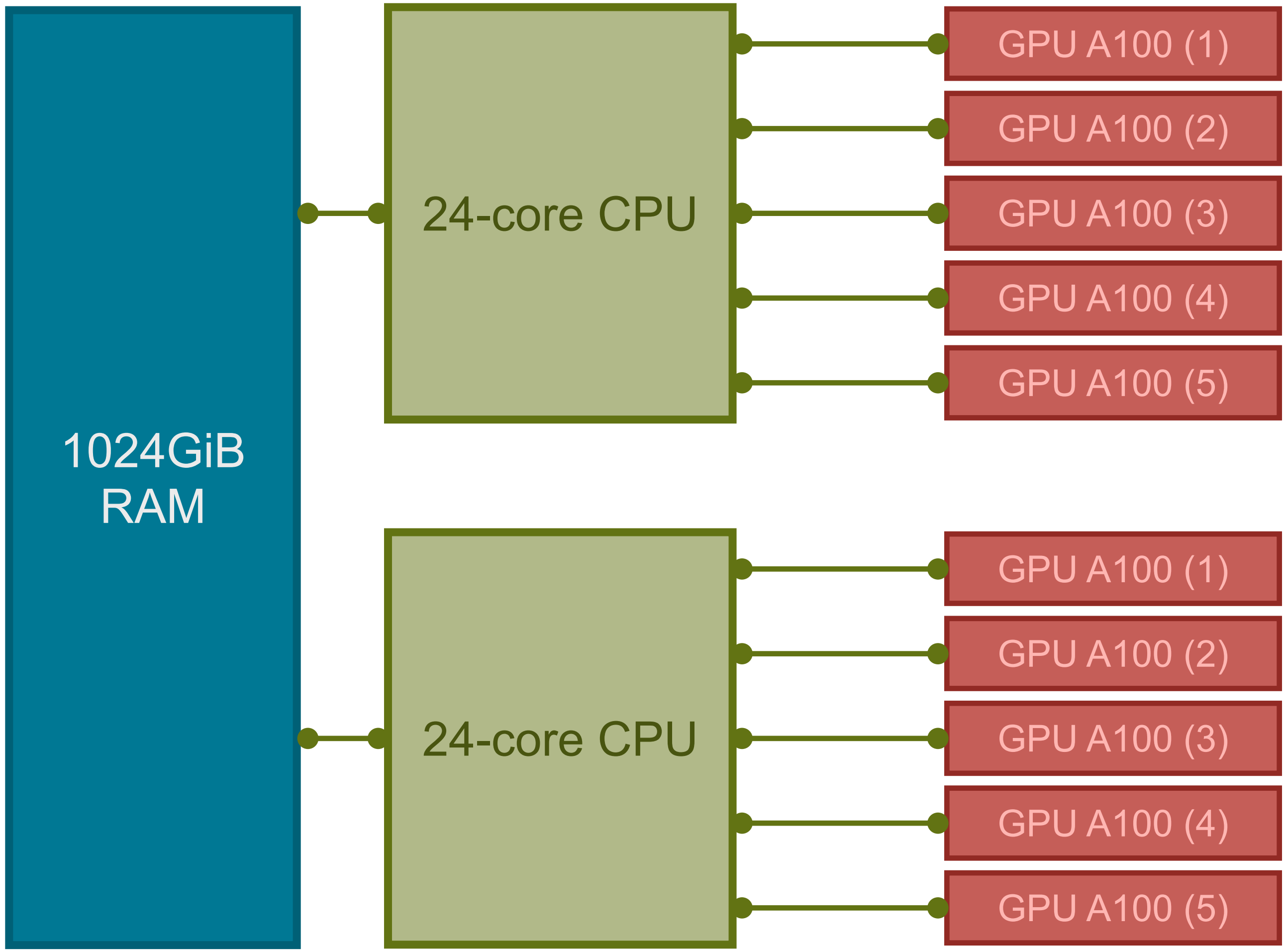


By requesting 1T of RAM, one will block the entire node, independently of the amount of GPUs requested at the same time

Same goes if one request all 48 cores

The rest of resources will be idle and unaccessible to other users

A very common pitfall in resource requirement :



By requesting 1T of RAM, one will block the entire node, independently of the amount of GPUs requested at the same time

Same goes if one request all 48 cores

The rest of resources will be idle and unaccessible to other users

Typically, AlphaFold2 users had no idea on which resources to request to fold a given protein, and why sometimes they were facing long pending times with SLURM

Knowing that my protein is N amino acids long, how long would it take to fold it ?

Which resources on Euler should I request ?

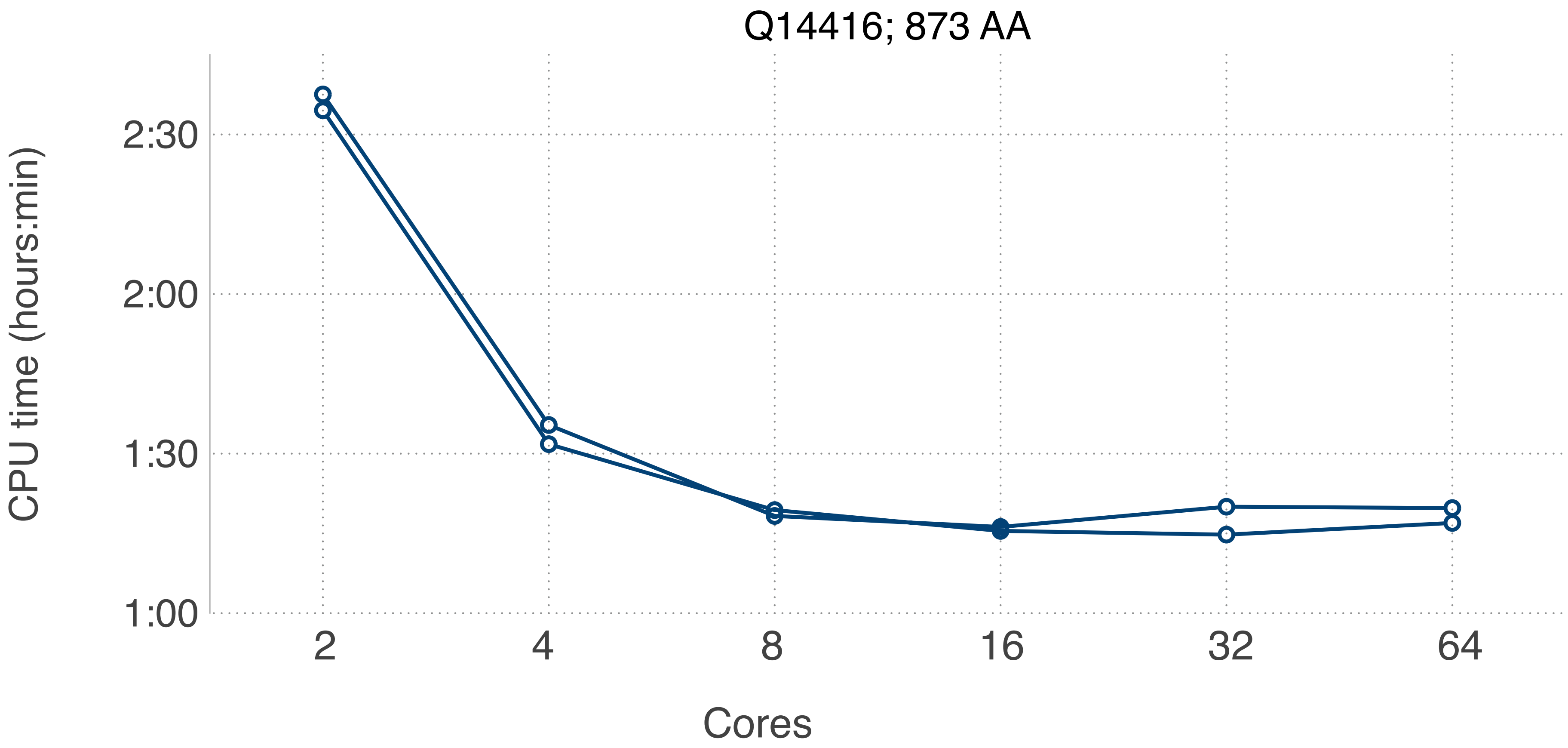
Benchmarking AF2 on the Euler cluster:

- All benchmarks has been done with version 2.3.1 of Alphafold

- We chose the following options :
  - Full databases queries
  - Amber relaxation
  - Relaxation and folding on GPUs (CPU-only option has also been explored)

- Monomer and multimer modes has been explored
- Protein lengths from 80 to 4900 AA
- Single GPU requested
- Fasta files of the proteins used for this benchmark are available here : https://gitlab.ethz.ch/sis/fastafiles_for_af2_tests

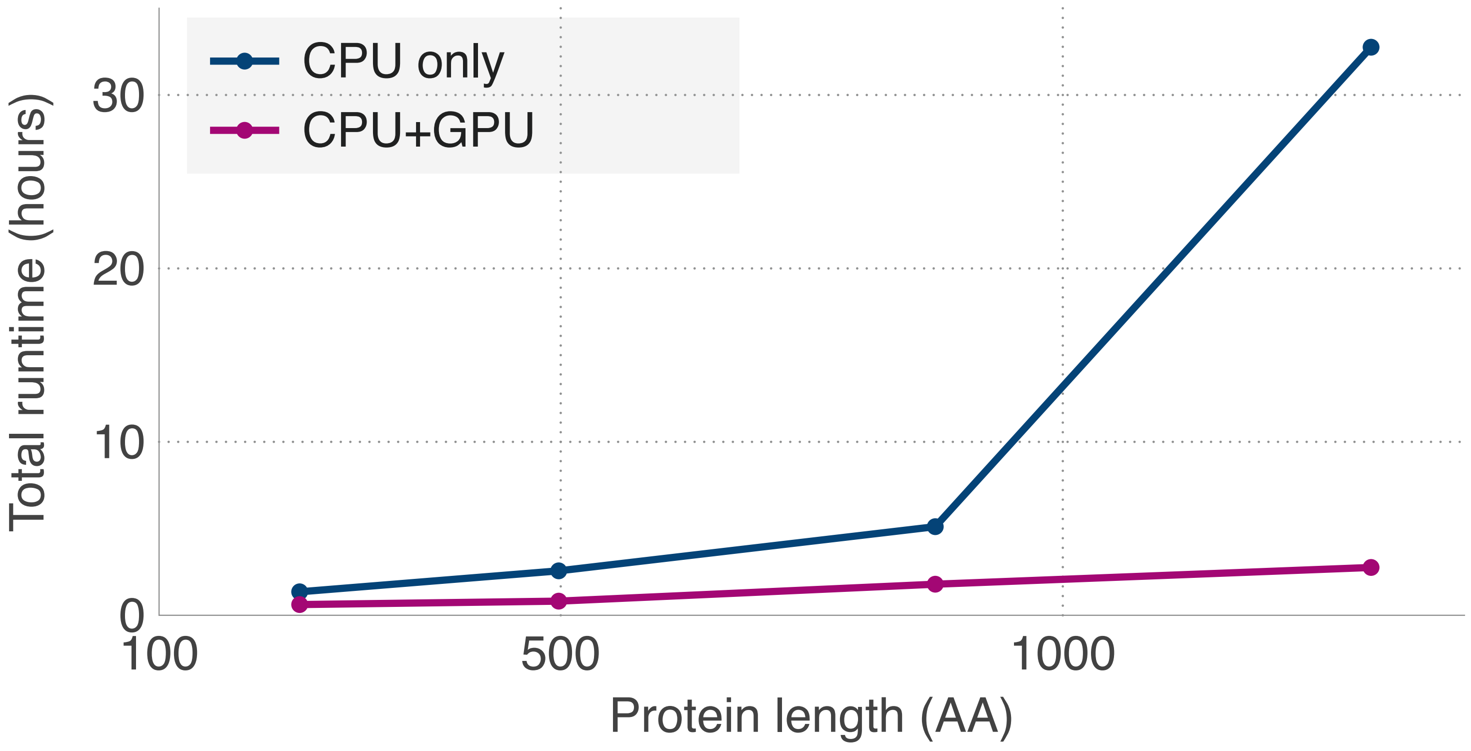
Benchmarking AF2 on Euler: parallel scaling



- **It is a waste of resources to request more than ~CPU 8 cores to run Alphafold**
- This result stands for the monomers and multimers

ETH zürich

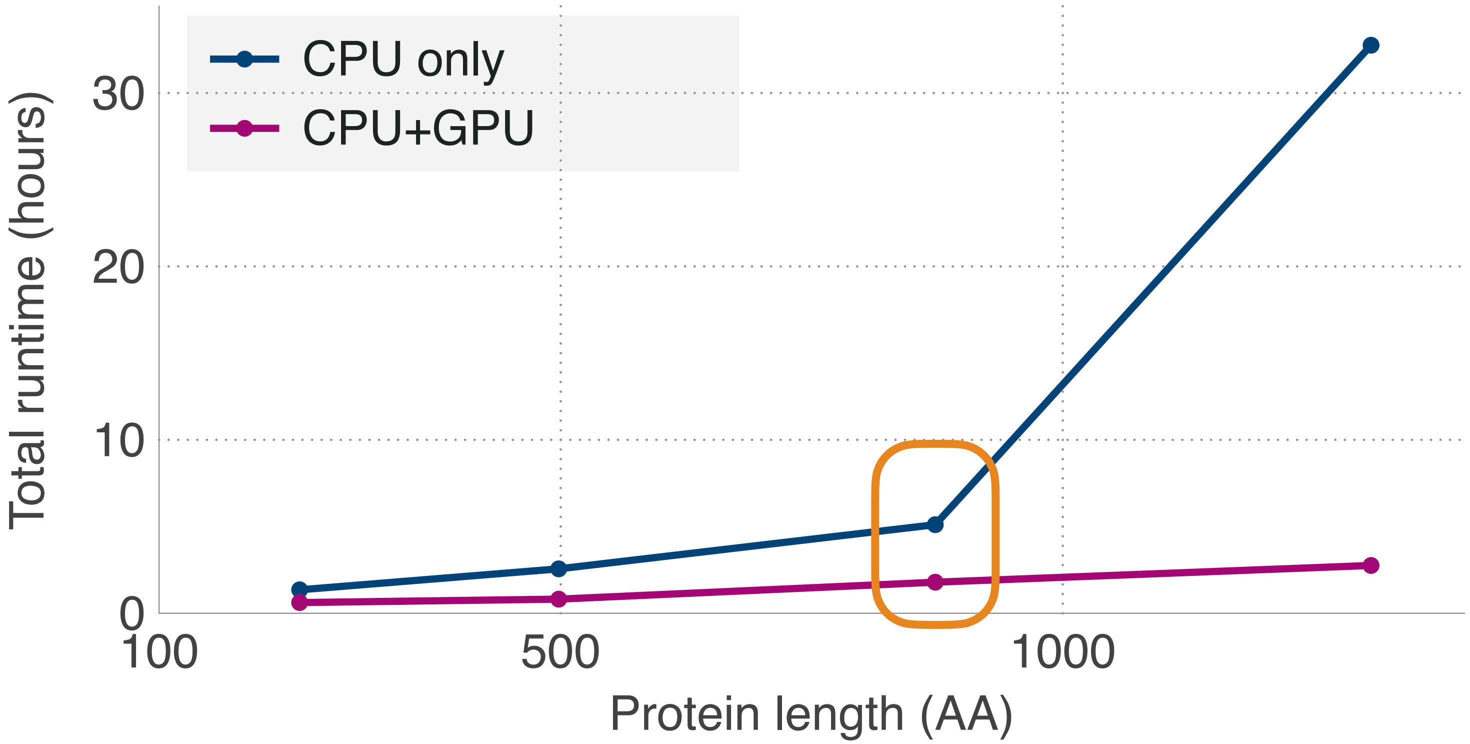
Benchmarking AF2 on Euler: CPU vs GPU



- As expected, the best time performance is obtained with GPUs
- Recommendation: fold only very small proteins (< 500 AA) only on CPU

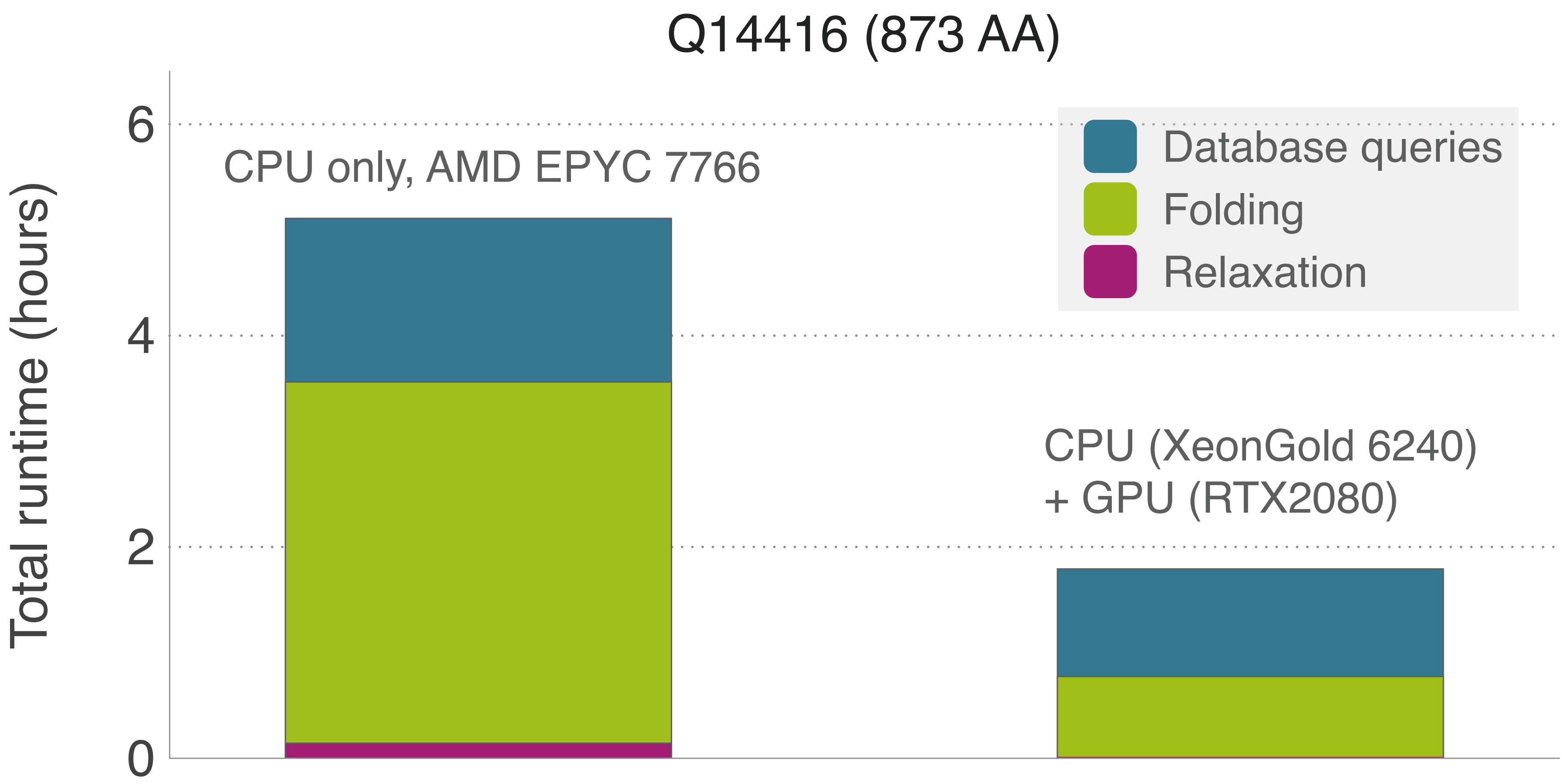
ETH zürich

Benchmarking AF2 on Euler: CPU vs GPU



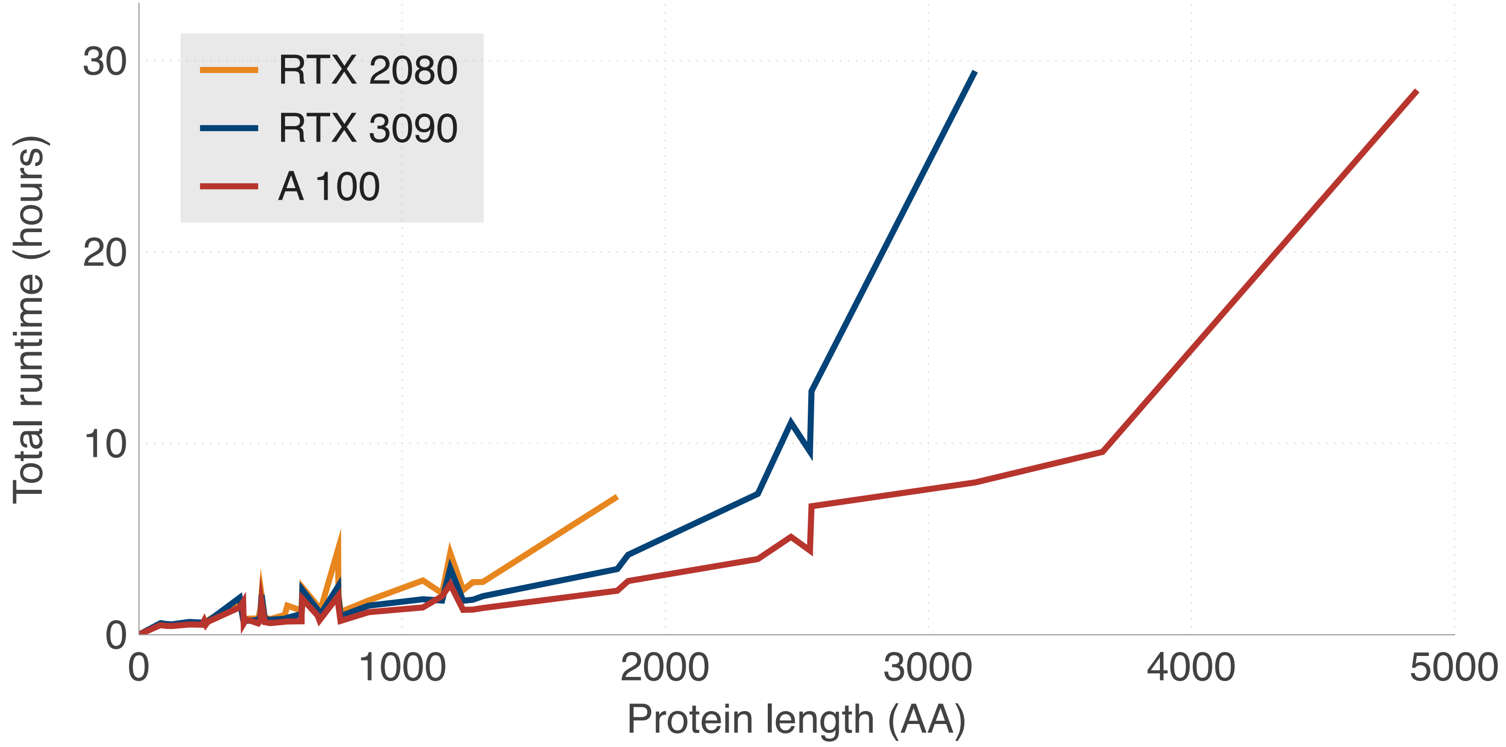
- As expected, the best time performance is obtained with GPUs
- Recommendation: fold only very small proteins (< 500 AA) only on CPU

Benchmarking AF2 on Euler: CPU vs GPU



- Performance improvement is due to a faster folding step on GPU

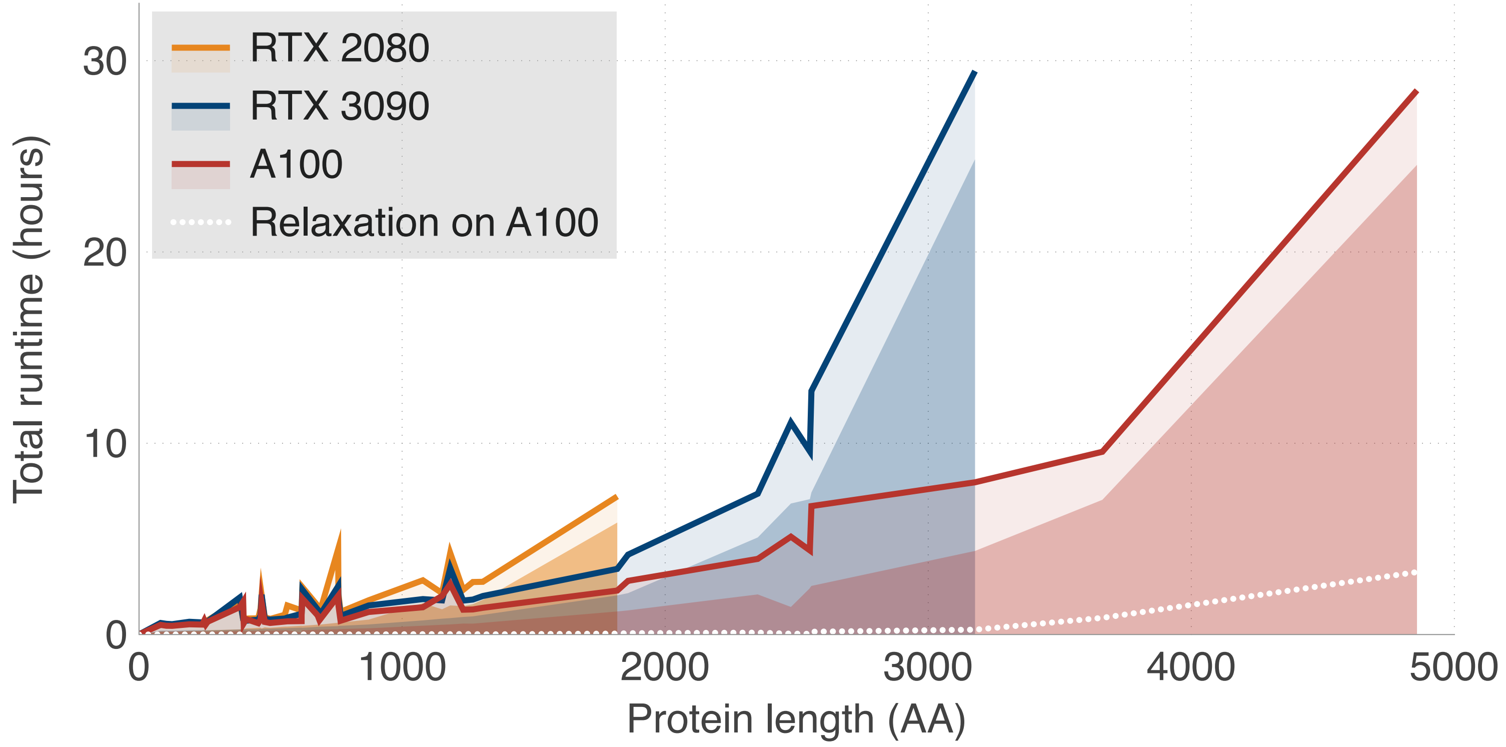
Benchmarking AF2 on Euler: length of protein sequence, monomer



- As expected, A100 professional GPUs are the best performing on the cluster
- RTX 2080 can fold proteins under ~1500 AA, loss of performance between 1000-1500AA
- RAM requirements are limited to ~210GB for monomers

ETH zürich

Benchmarking AF2 on Euler: length of protein sequence, monomer



- The difference in performance is clearly due to the GPU performance (lower shaded part of each curve)
- Time devoted to the database queries is roughly similar and will not be the main factor influencing the job time for proteins >1500 AA

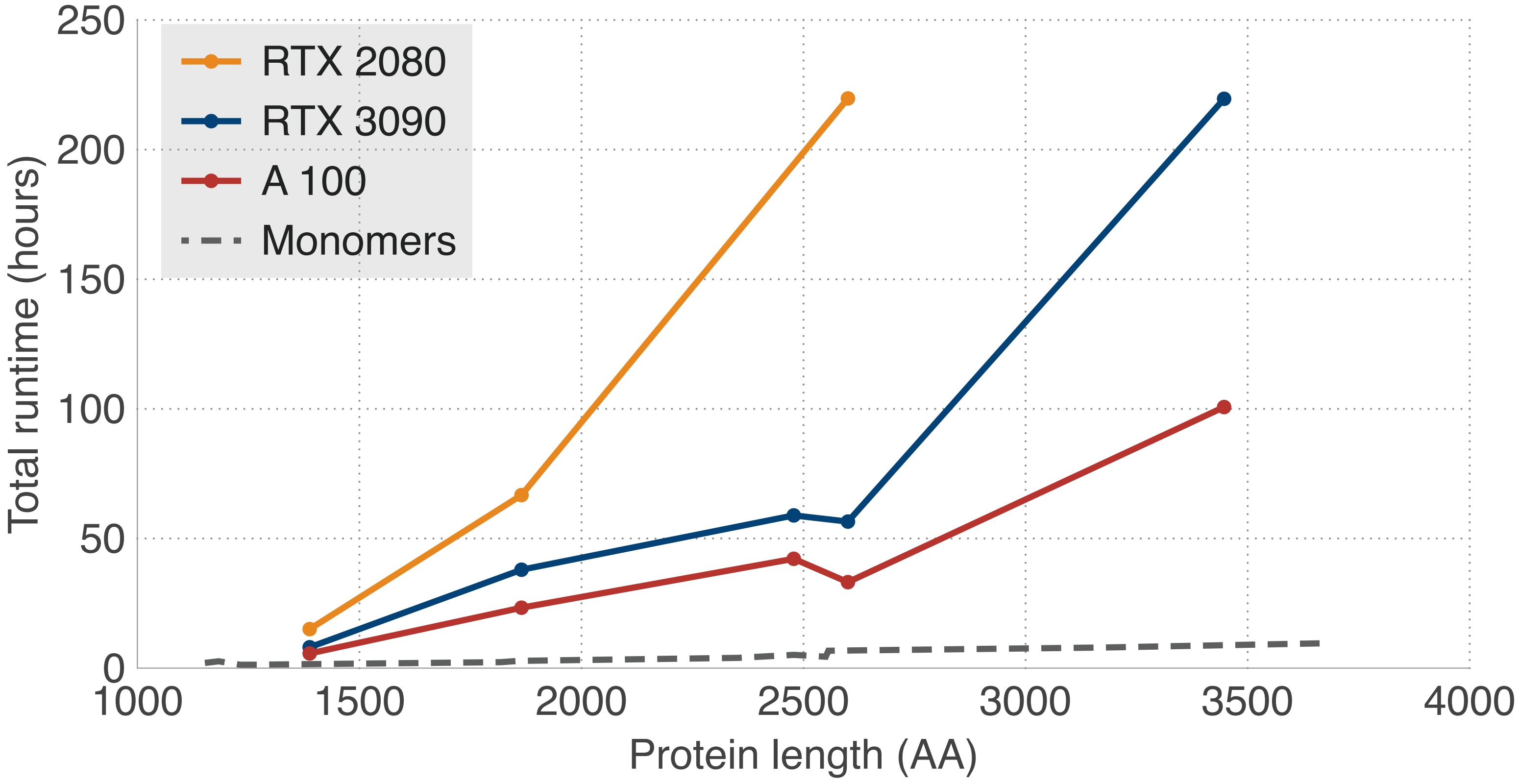
Benchmarking AF2 on Euler: length of protein sequence, monomer

For monomers :

- request 8 CPU cores, at most 230GB of RAM

Monomer size	Recommended GPU model
< 500 AA	Gamer GPU : NVIDIA GeForce RTX 2080 Ti
> 500 AA **< 2500 AA**	Intermediate to high-range GPUs : NVIDIA GeForce RTX 3090, NVIDIA TITAN RTX, NVIDIA Tesla V100-SXM2 32 GiB
> 2500 AA	Professional GPUs : Nvidia Tesla A100 (40 GiB), Nvidia Tesla A100 (80 GiB)

Benchmarking AF2 on Euler: length of protein sequence, multimer



- A100 would be preferred for any multimer >2500 AA
- Multimer runs have huge CPU (and GPU) memory requirements, CPU RAM value is unconstrained

ETH zürich

Benchmarking AF2 on Euler: length of protein sequence, multimer

For multimers :

- request 8 CPU cores, choice of RAM has to be done by trial/error

Multimer size	Recommended GPU model
< 1000 AA	Gamer GPU : NVIDIA GeForce RTX 2080 Ti (Generally we do not recommend using single gamer-GPUs for multimer folding because of huge memory requirements)
> 1000 AA < 2500 AA	Intermediate to high-range GPUs : NVIDIA GeForce RTX 3090, NVIDIA TITAN RTX, NVIDIA Tesla V100-SXM2 32 GiB
> 2500 AA	Professional GPUs : Nvidia Tesla A100 (40 GiB), Nvidia Tesla A100 (80 GiB)

Conclusions:

- Very widely used ML codes could be developed without the consideration of their performance on an HPC cluster

- Potential issues could include :
  - Idle resources, such as CPU/GPU parts of the code (while CPU is running, GPU is idle or vice-versa)
  - Over-request of resources, leading to excessively long pending times
  - Hard-encoded variables and paths that can limit the performance, complicate the installation

- For AF2 to mitigate these issues we have created a SLURM script generator that enforces our recommendations for optimal resource usage on Euler, but also allows the user to adjust resources requirements if needed

Thank you !