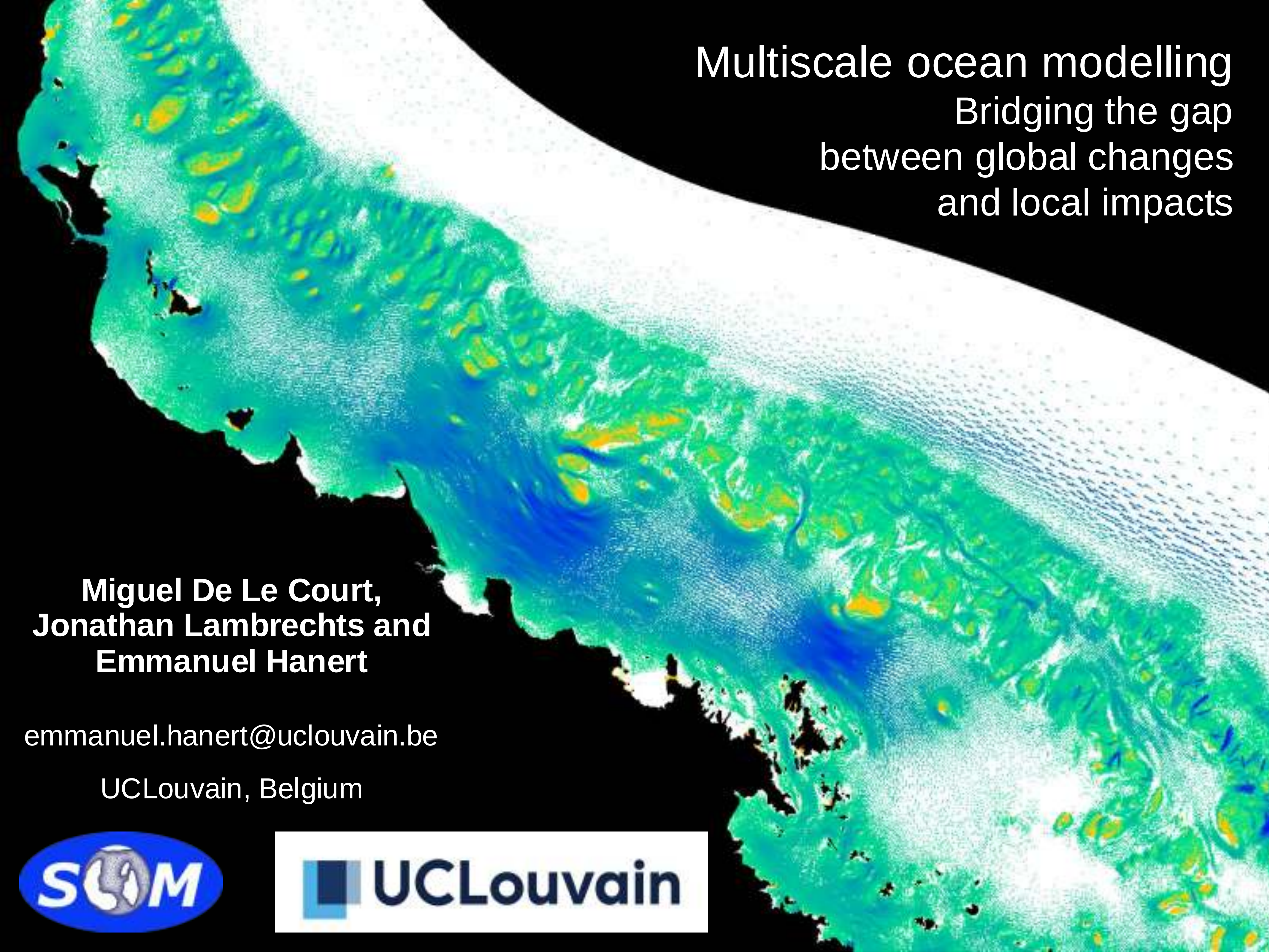


Multiscale ocean modelling

Bridging the gap between global changes and local impacts

Miguel De Le Court, Jonathan Lambrechts and Emmanuel Hanert

emmanuel.hanert@uclouvain.be

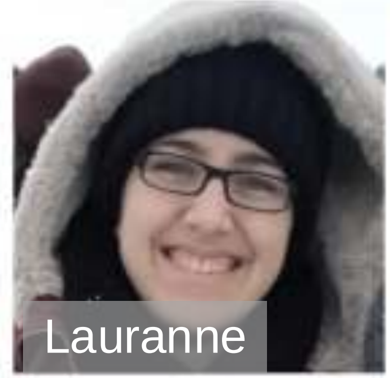
UCLouvain, Belgium

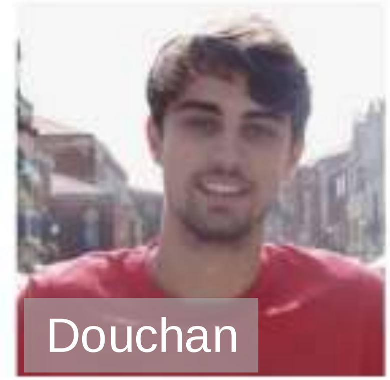
SLIM

UCLouvain

SLIM people

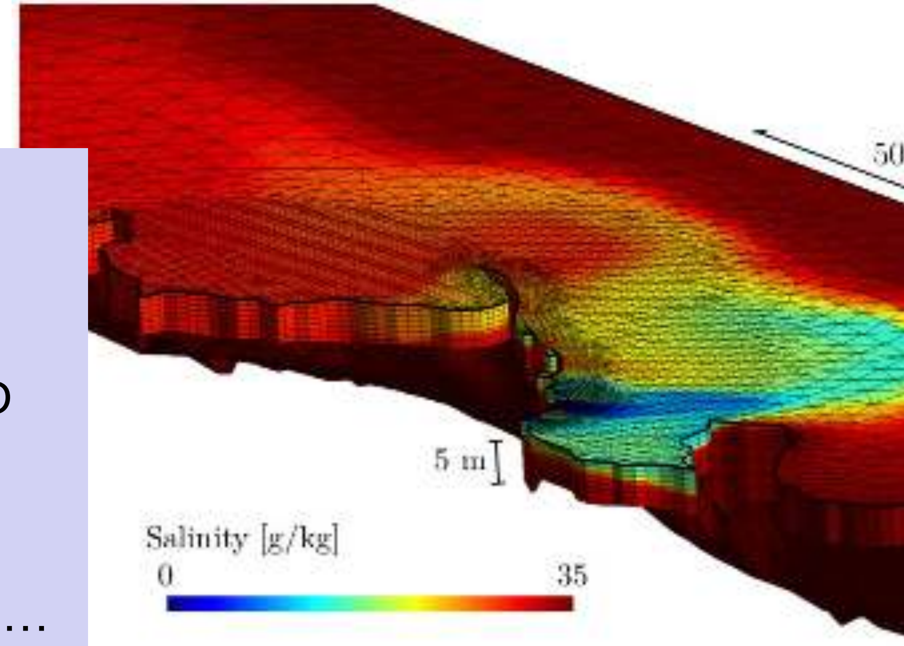
- 2 academics
- 2 post-doc researchers
- 11 PhD researchers
- www.slim-ocean.be

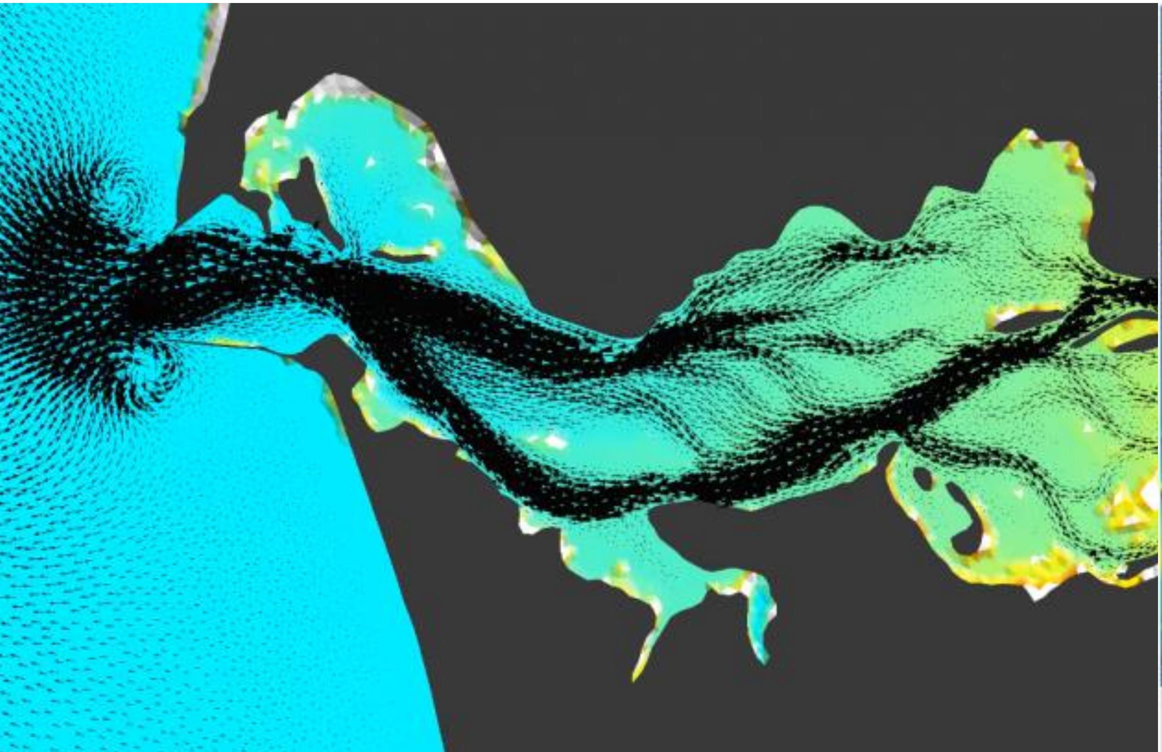


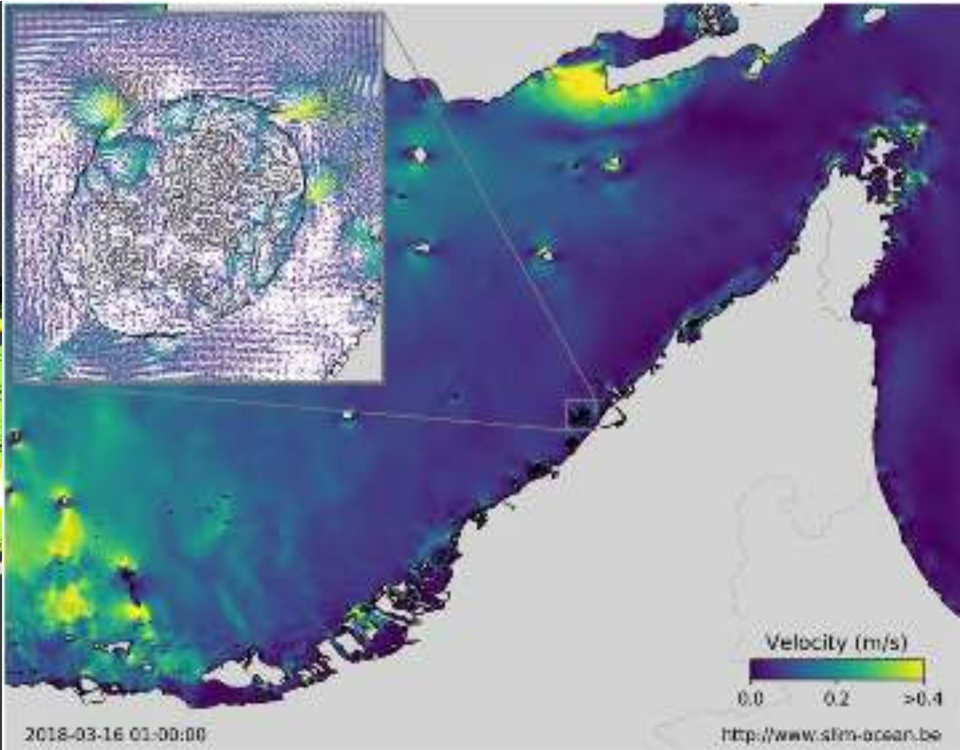


SLIM in a nutshell…

- Unstructured meshes → **variable resolution**
- Discontinuous Galerkin finite element method
- Different **hydrodynamic models**
    - SLIM1D for river flows
    - SLIM2D for shallow barotropic flows with W&D
    - SLIM3D for hydrostatic baroclinic flows
- Coupled with the coastal **wave model** SWAN
- Coupled with different **transport models**
    - Eulerian: sediments, water age, water quality, …
    - Lagrangian: coral larvae, plastic debris, seagrass propagules, turtle hatchlings, oil spills, …
- Hydrodynamic models run on **GPUs**.





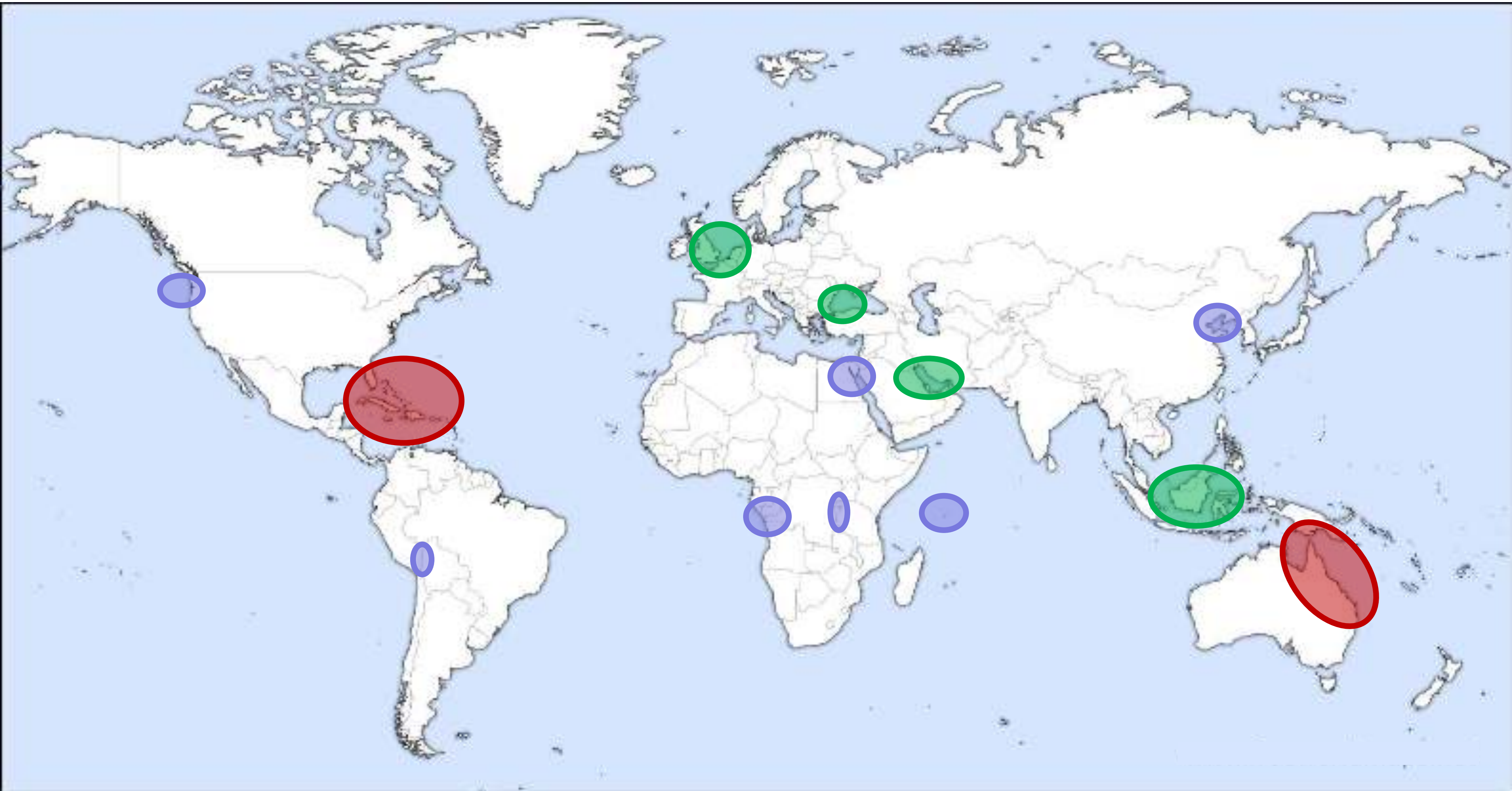


SLIM in the world…

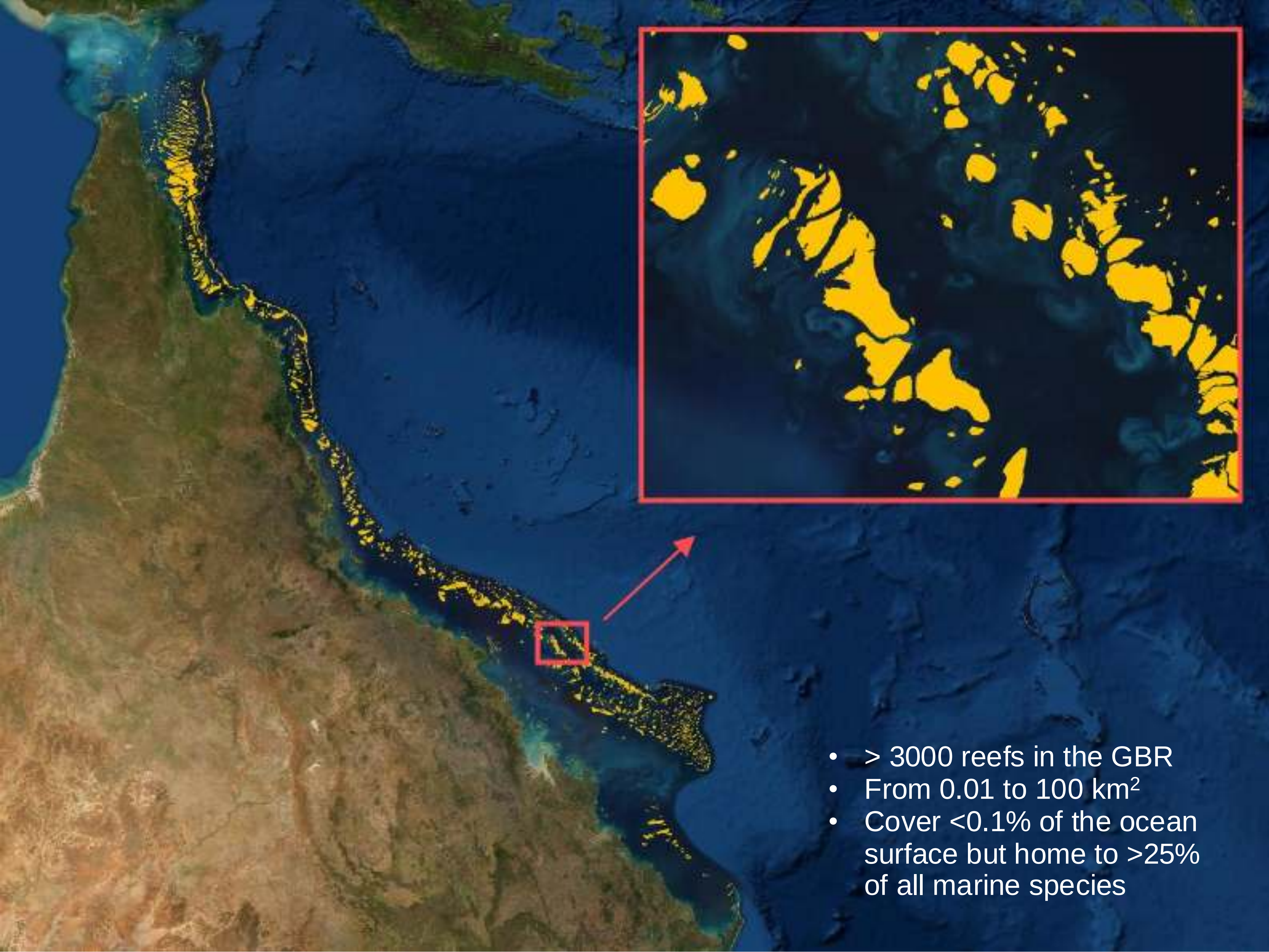
- Great Barrier Reef, Australia
- Florida Reef Tract, Bahamas, USVI

- Scheldt, North Sea
- Danube, Black Sea
- Arabian/Persian Gulf
- Indonesia

- Columbia River
- Lake Titicaca
- Congo River
- Lake Tanganyika
- Seychelles
- Northern Red Sea
- Bohai Sea

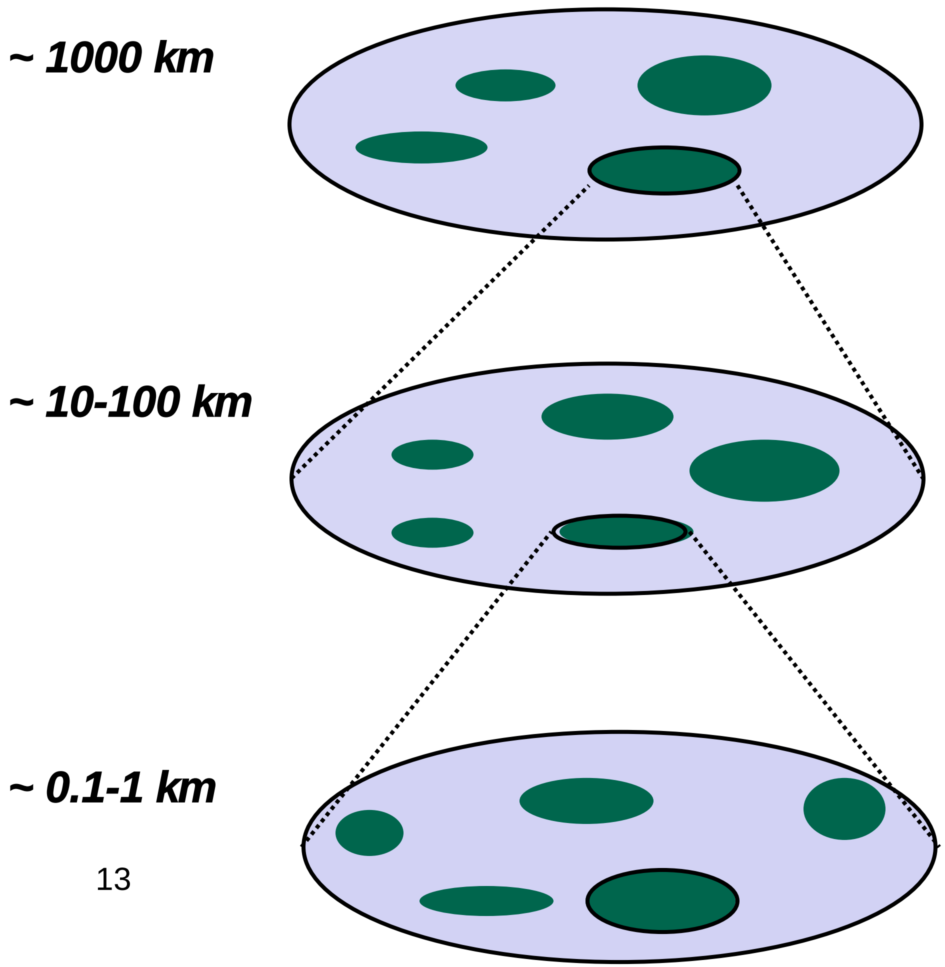



2000 km



- > 3000 reefs in the GBR
- From 0.01 to 100 km^2
- Cover <0.1% of the ocean surface but home to >25% of all marine species

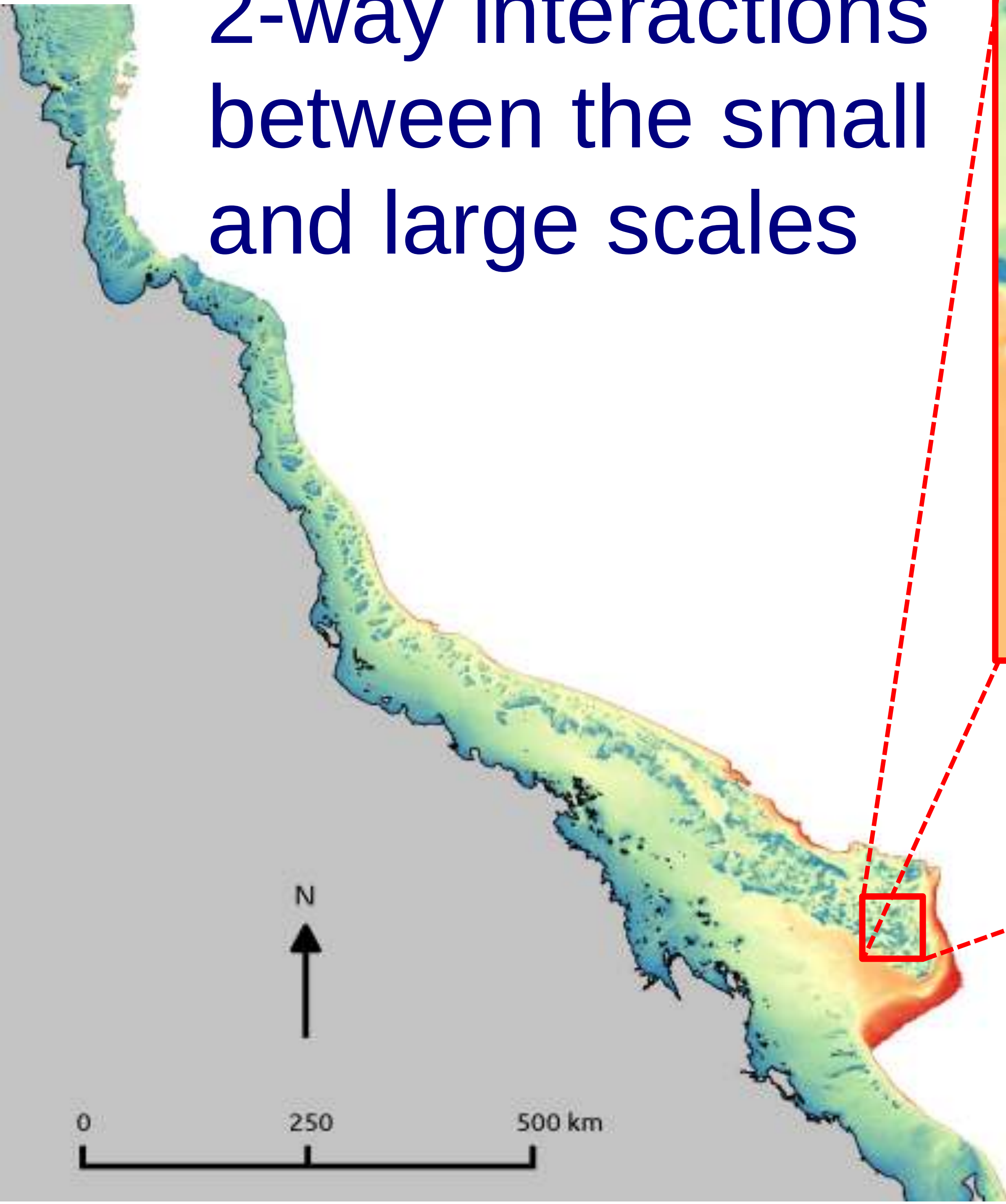
Like most marine environments, the GBR is a large multi-scale system

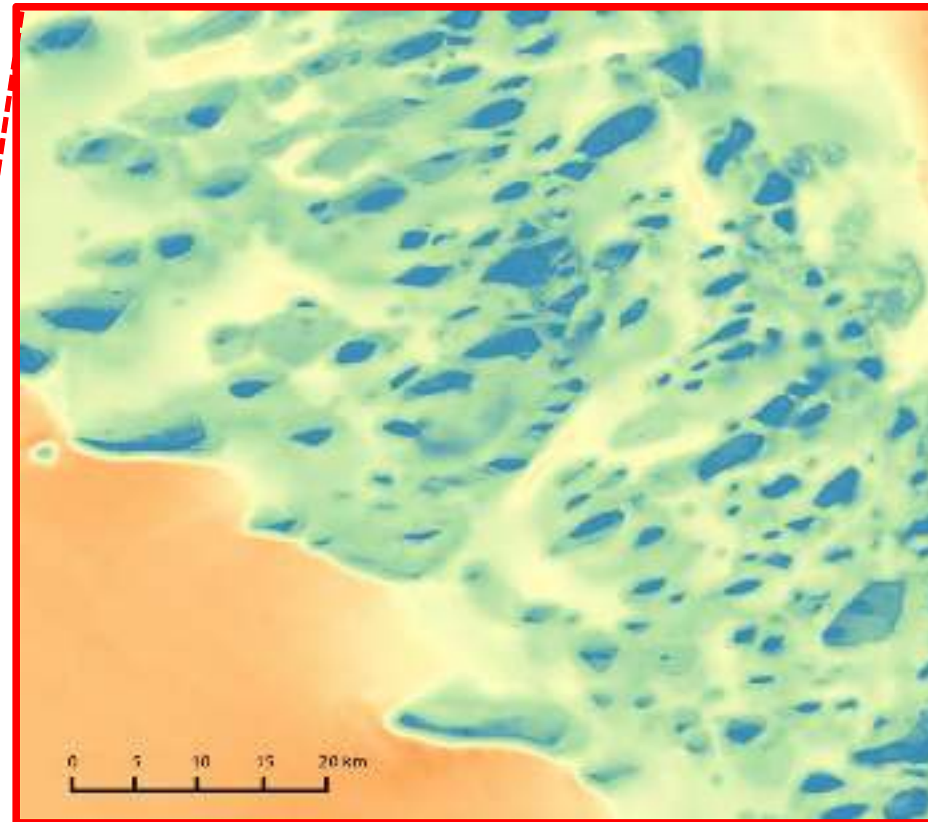


Management is done at the large scale (e.g. for MPAs and regulations)

Our understanding of the reef-building coral species is at the small scale

2-way interactions between the small and large scales

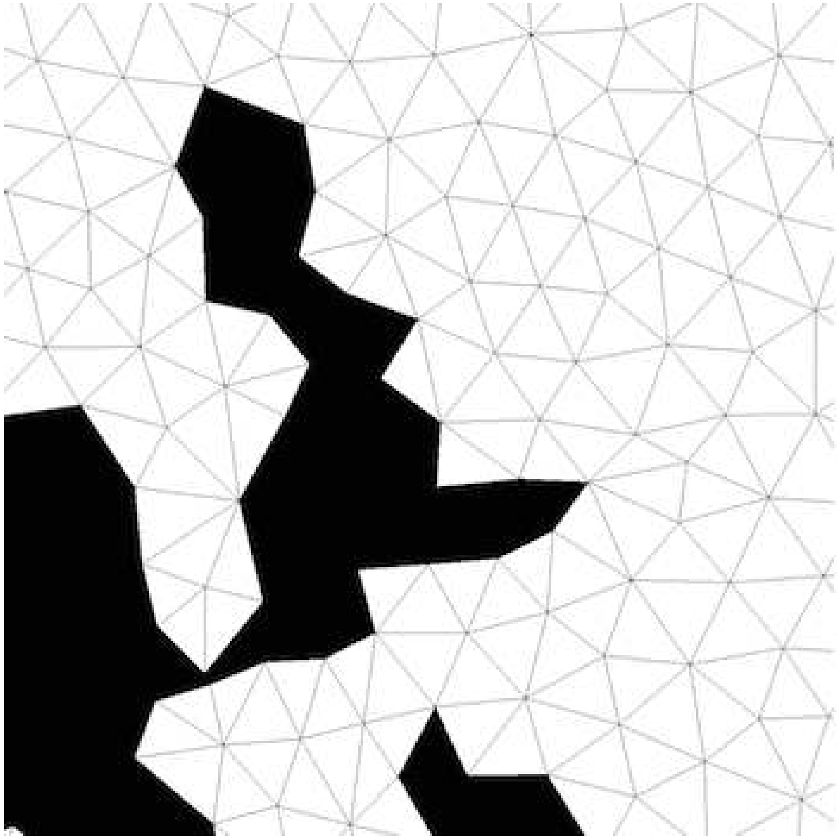


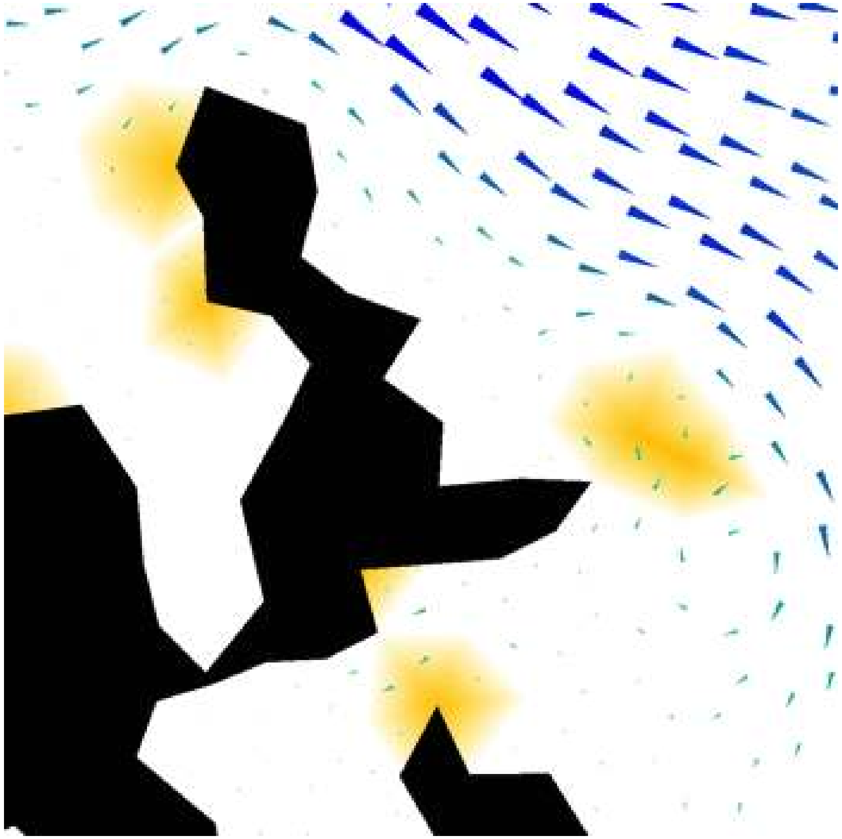


Different processes, at different scales, interact together

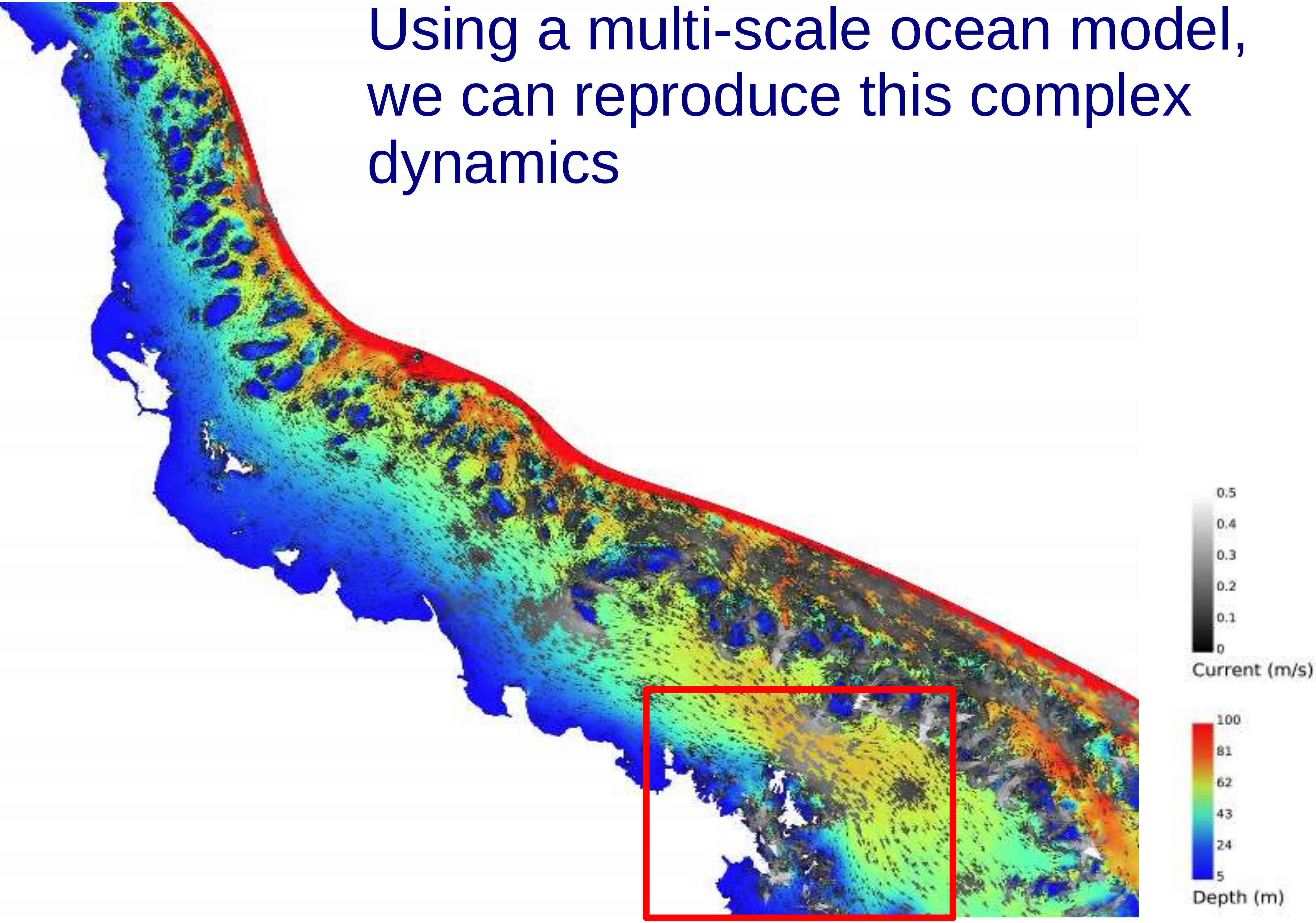
- ***The large-scale currents from the Pacific are deflected by the reefs.***
- ***The small-scale flow features around the reefs, like eddies, are driven by the interplay between the tides, the wind and the large-scale currents.***
- ***A model of the GBR should represent these different processes and their interactions from*** *1000's km to 10's m.*

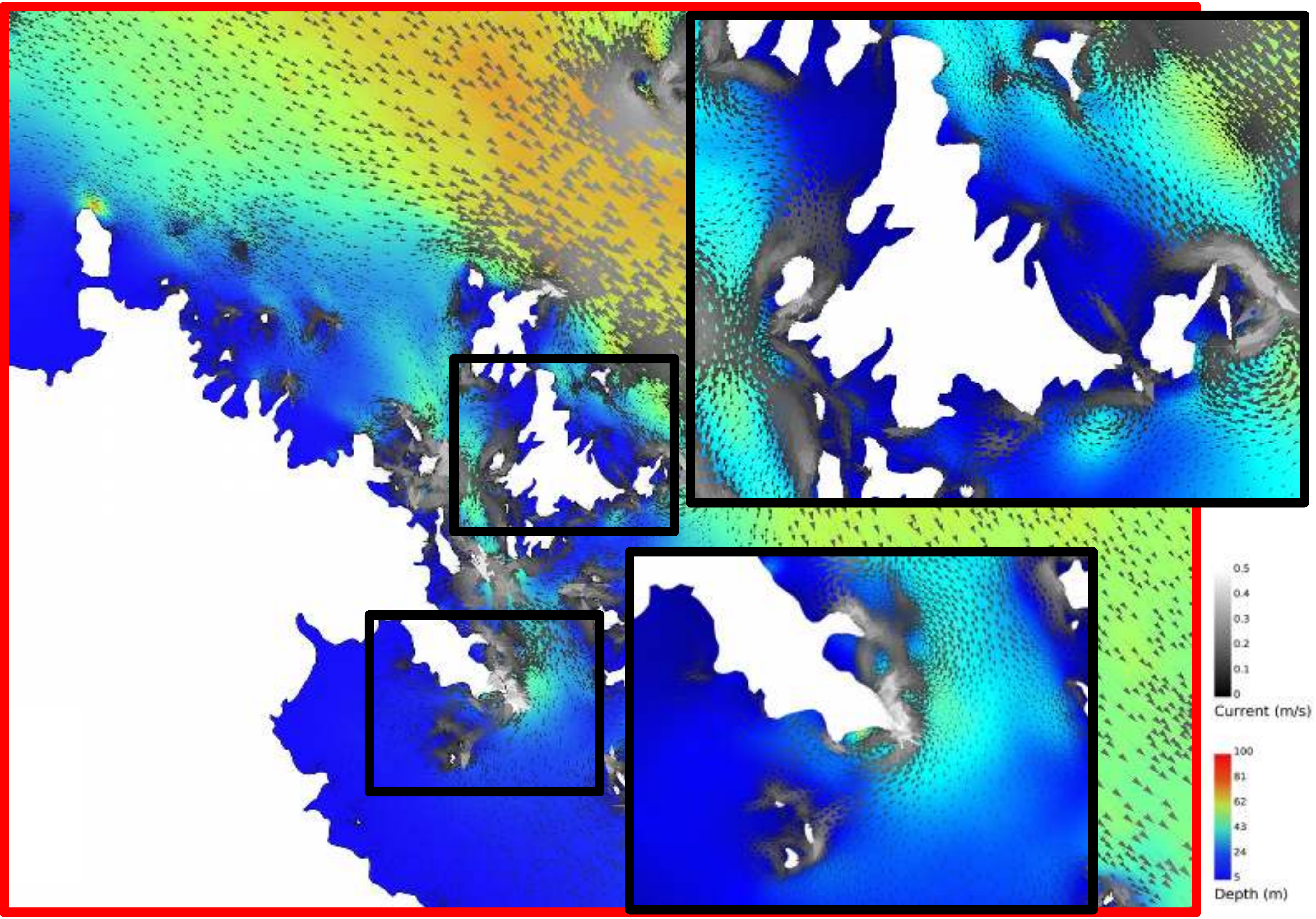
Getting the right resolution, at the right place is really important…





Using a multi-scale ocean model, we can reproduce this complex dynamics





0.5
0.4
0.3
0.2
0.1
0
Current (m/s)
100
81
62
43
24
5
Depth (m)

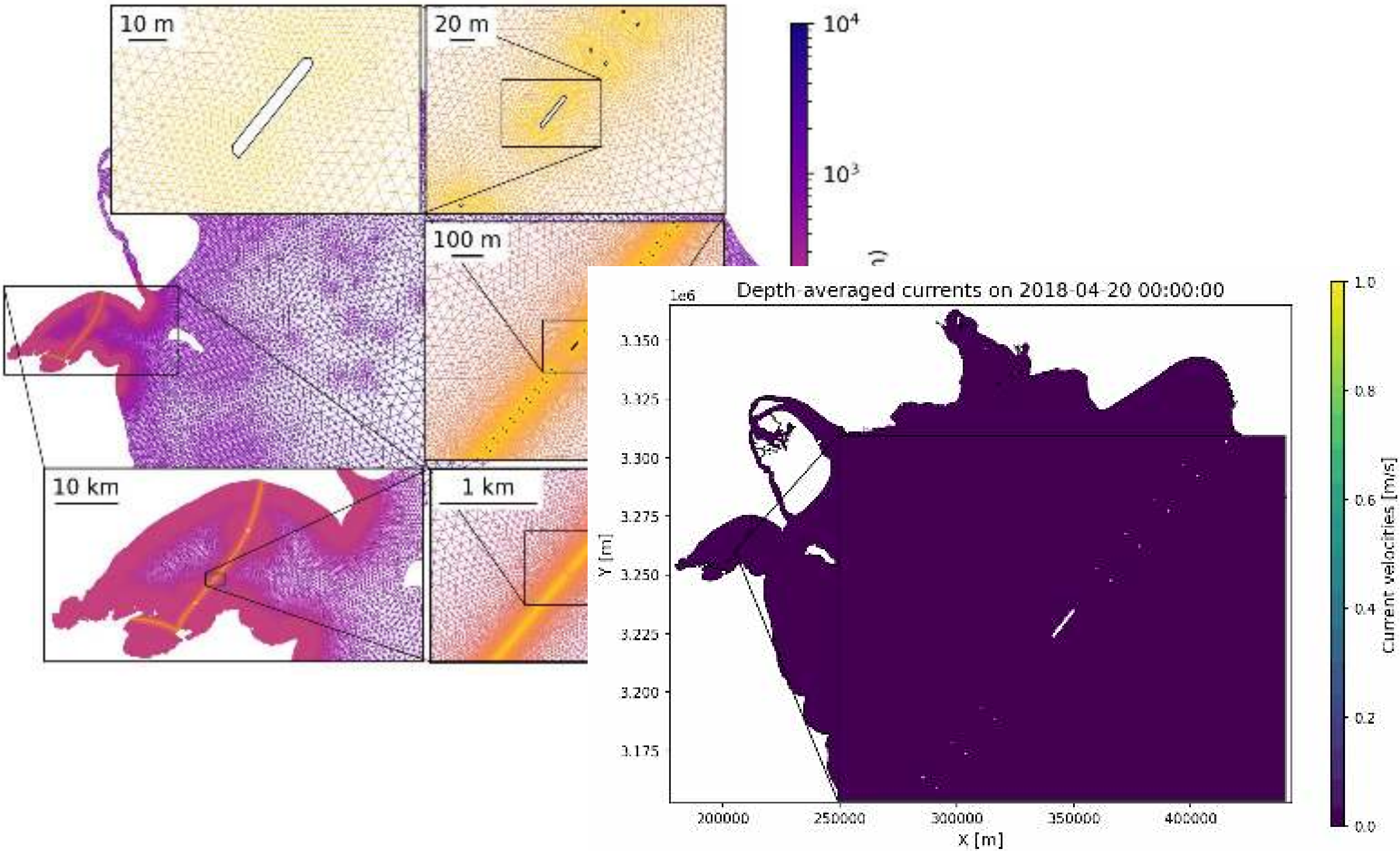
The same applies to artificial structures instead of reefs

Sheikh Jaber & Doha Link
36km + 12km
Causeways

Artificial islands
2x 0.28 km^2
Reclaimed area

Shuwaikh Port
0.38 km^2
Reclaimed area

The multiscale nature of the flow again requires a multiscale model



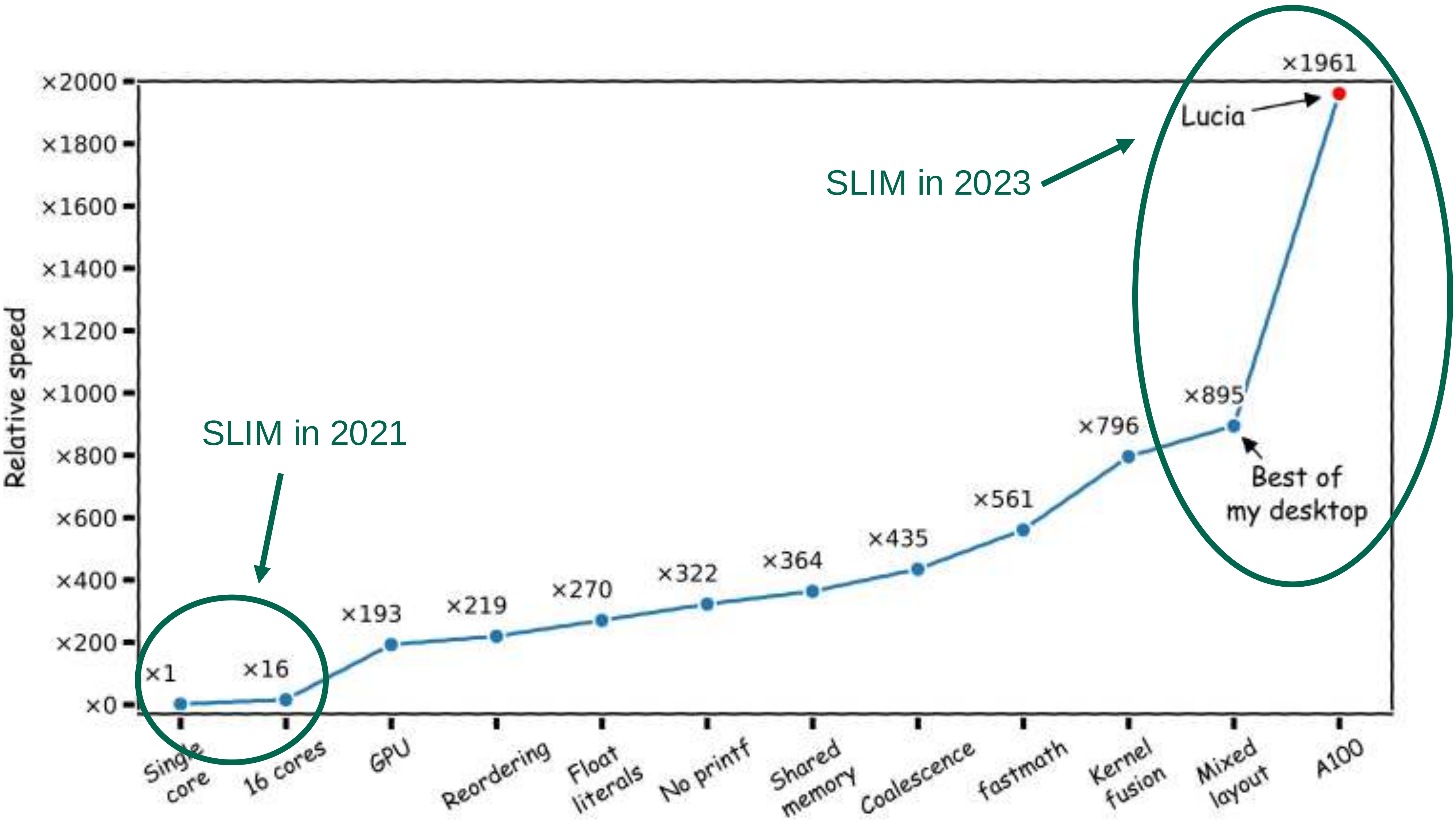
Increasing resolution is however not enough…

- As we increase the resolution, we have to reconsider the **physical assumptions** underlying our model.
- A 2D depth-averaged model is often not sufficient to capture the dynamics of marine flows, which are often driven by density gradients and hence **3D**.
- As the mesh resolution becomes of the order of water depth, **non-hydrostatic** effects can no longer be neglected neither.

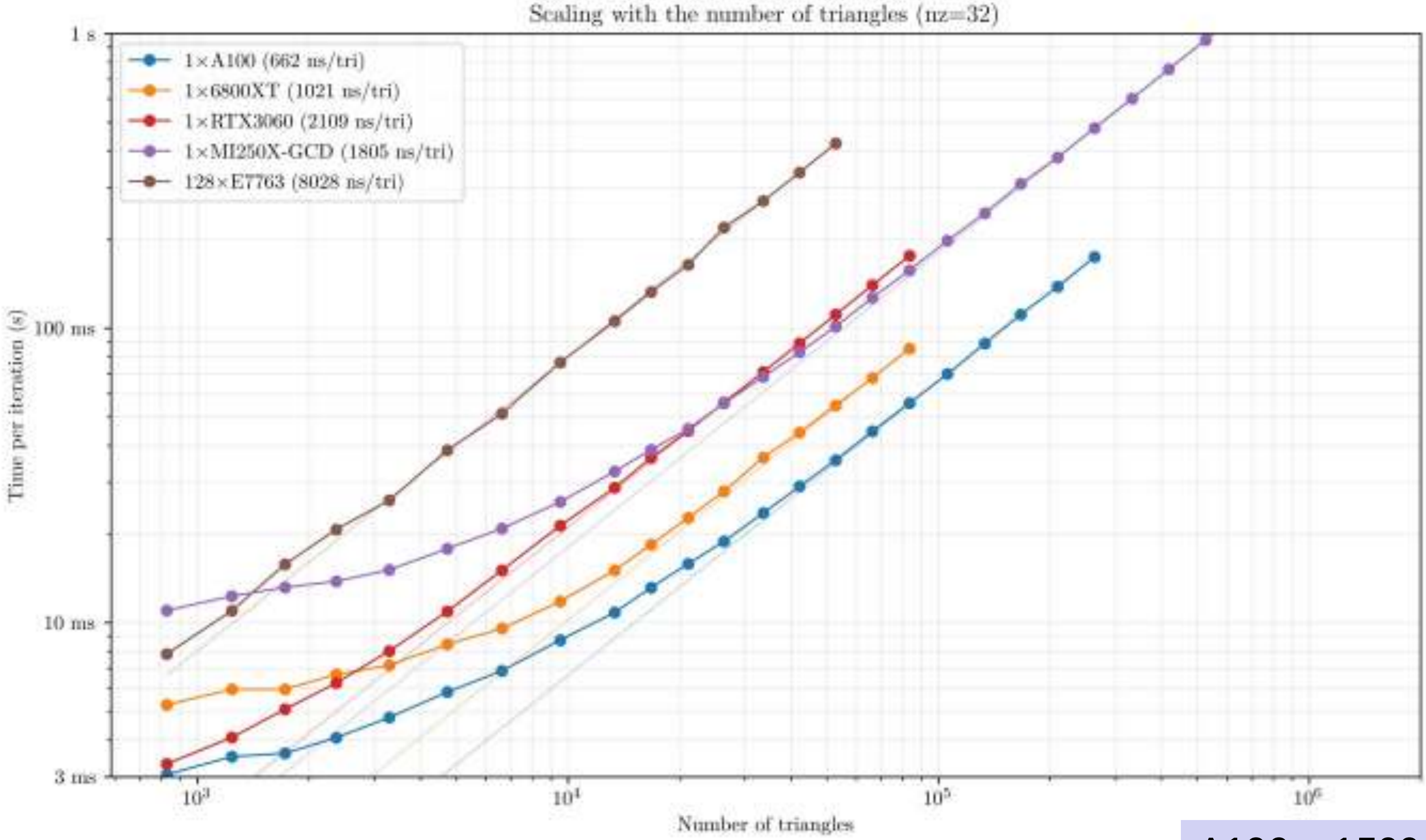
$\Rightarrow$ We hence have to consider models with a more complex physics

$\Rightarrow$ Move from a multi-scale to a **multi-physics** modeling paradigm.

To do that, we had to completely rewrite our model to run on GPUs…



Single-GPU performance of the 3D model



A100 = 1500 cores
6800XT = 1000 cores
MI250X-GCD = 550 cores
RTX3060 = 480 cores

Scaling to multiple GPUs

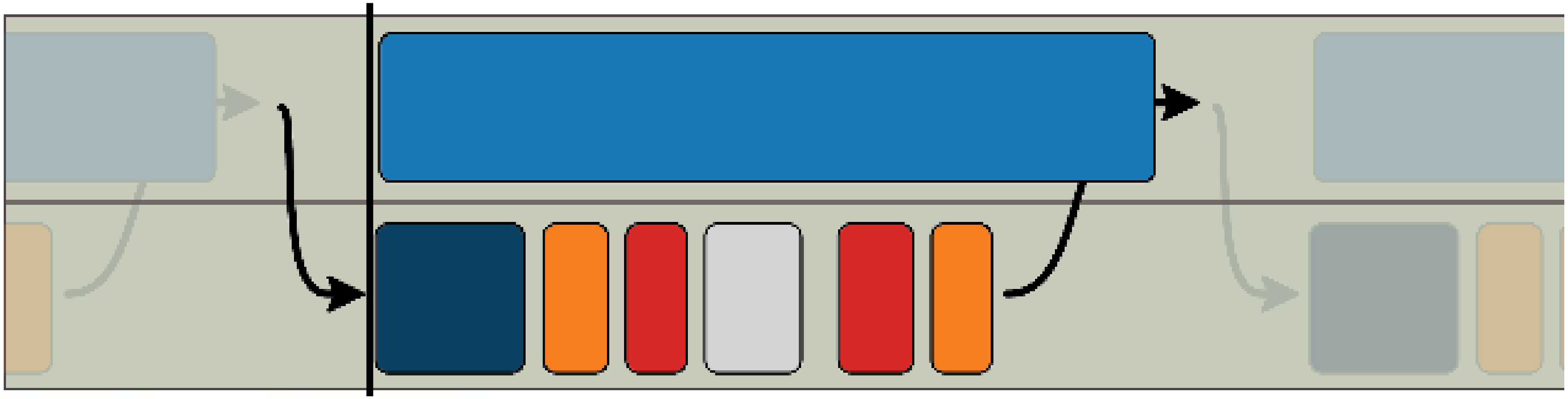
Naïve approach : Compute, then communicate

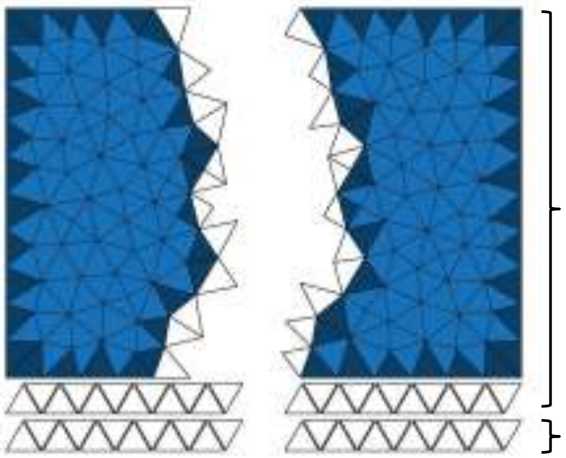
- Works very well on CPUs where the computation is very slow compared to communications
- Completely unusable with GPUs : The computation time is way too short

Better approach : Overlap communication and computation

- Requires splitting the computation : boundaries VS interior
- On GPUs, requires multiple 2 streams and explicit synchronization for good performance

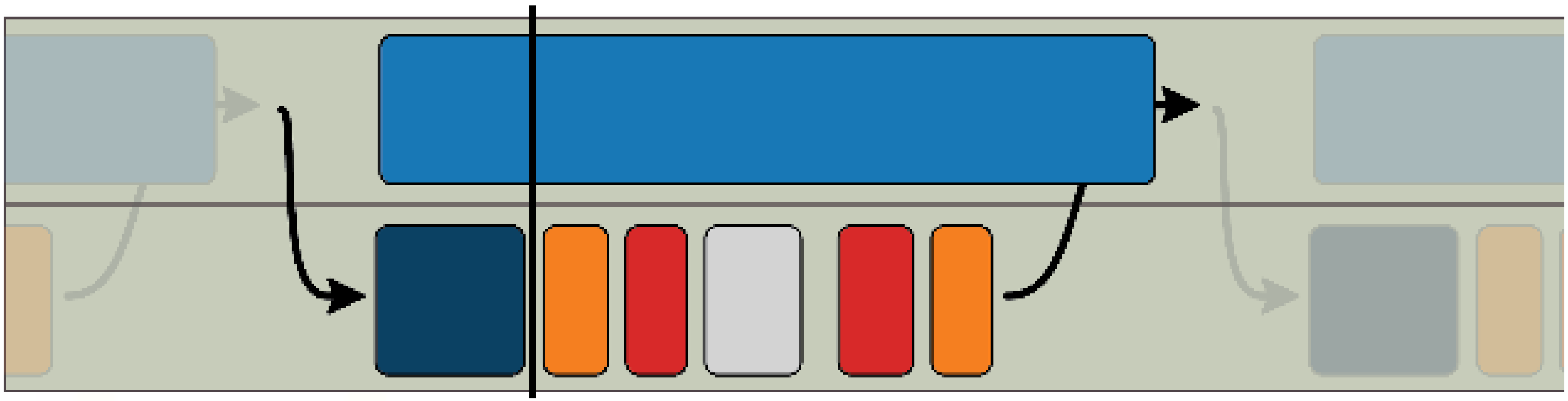
Structure of communications

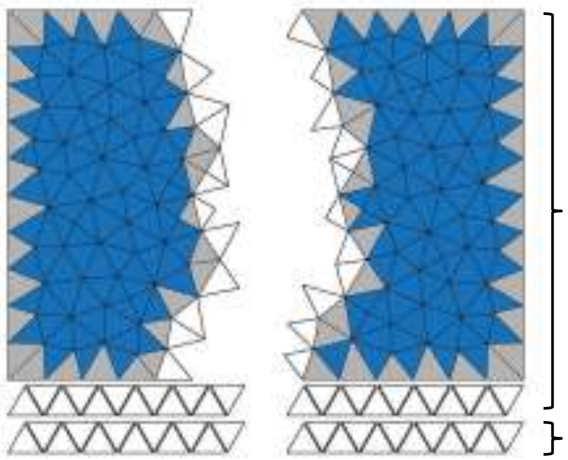




1. Launch the computation on the boundary

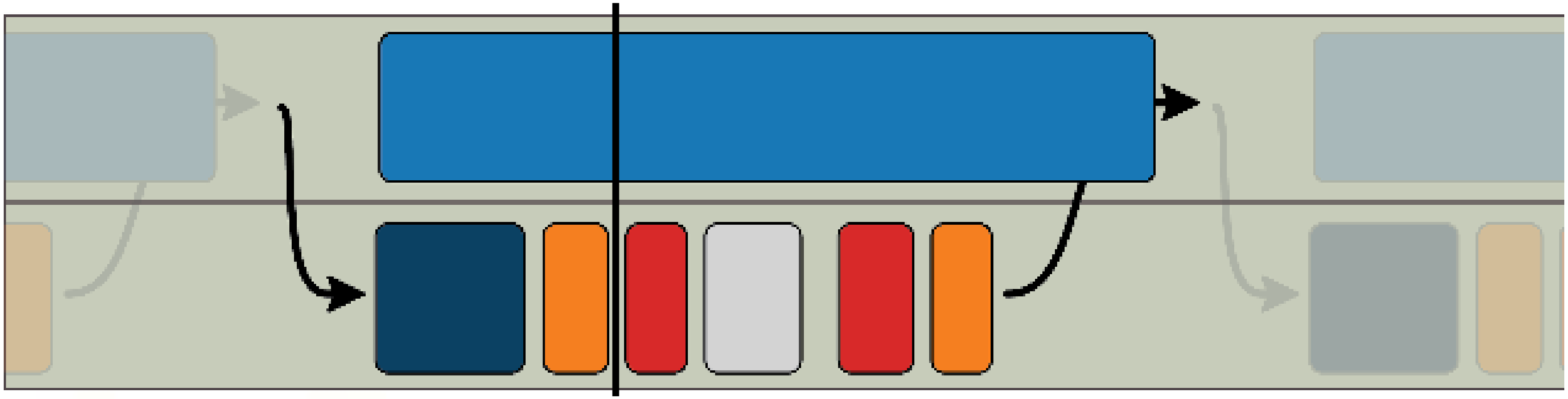
Structure of communications

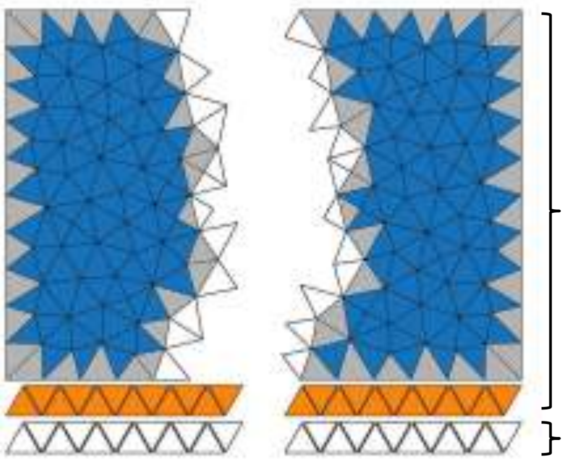




1. Launch the computation on the boundary
2. Pack the data that needs to be sent in a buffer

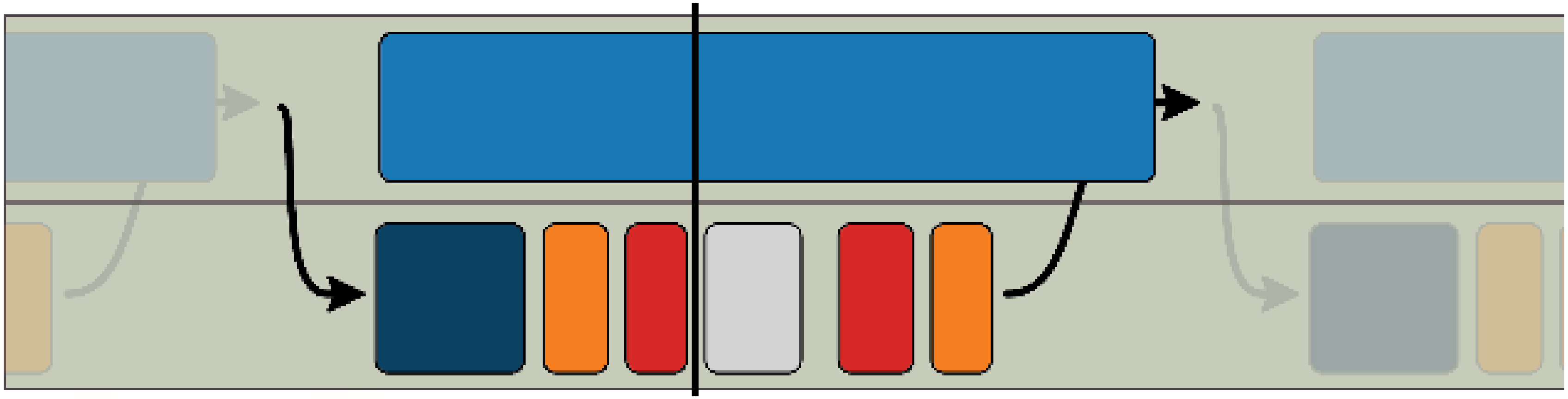
Structure of communications

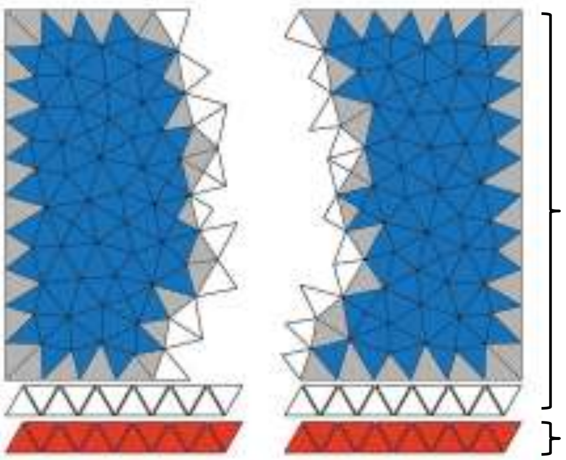




1. Launch the computation on the boundary
2. Pack the data that needs to be sent in a buffer
3. Copy to the CPU

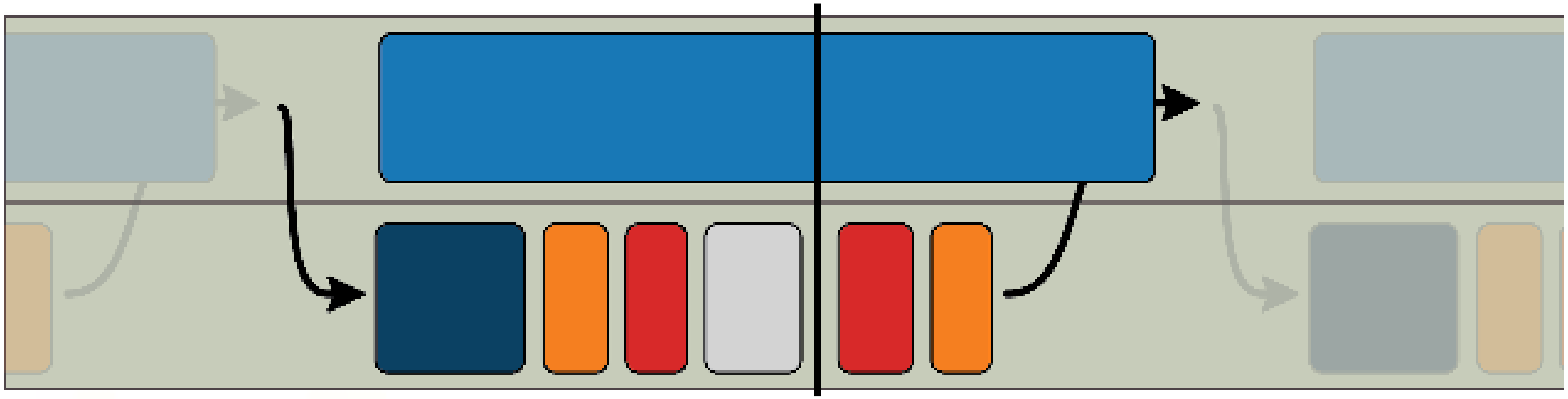
Structure of communications

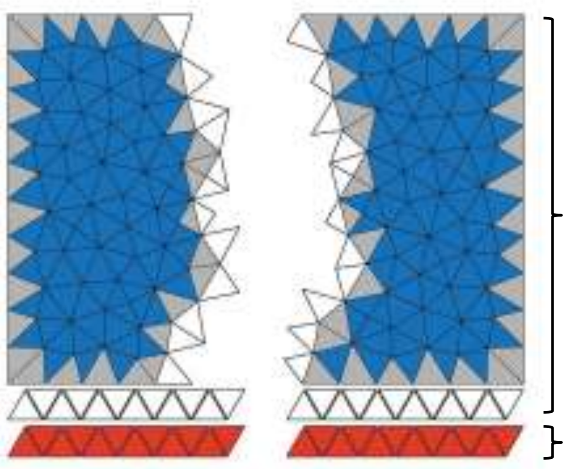




1. Launch the computation on the boundary
2. Pack the data that needs to be sent in a buffer
3. Copy to the CPU
4. Communicate with MPI

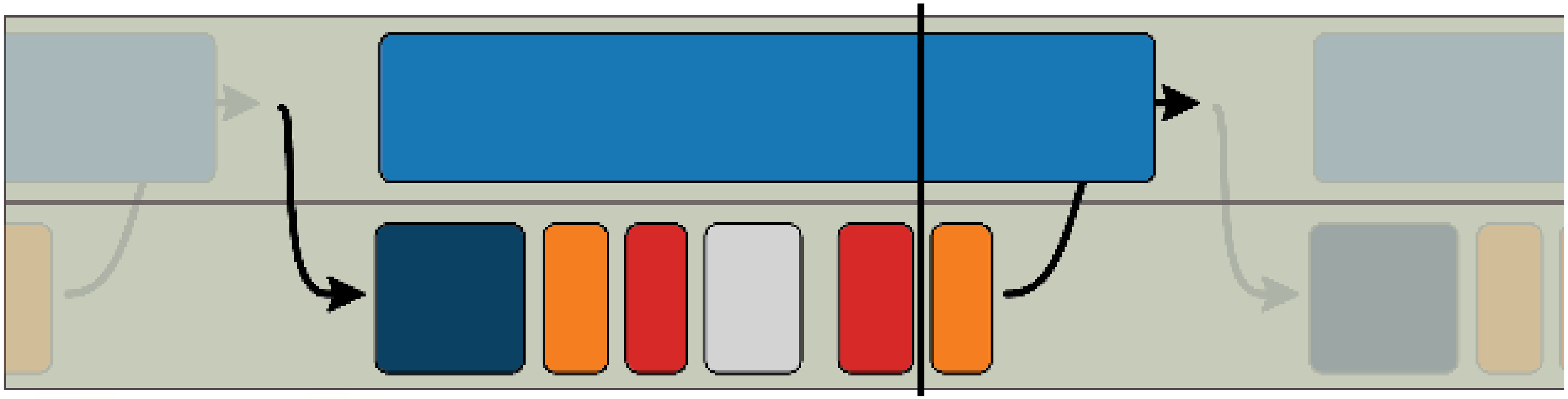
Structure of communications

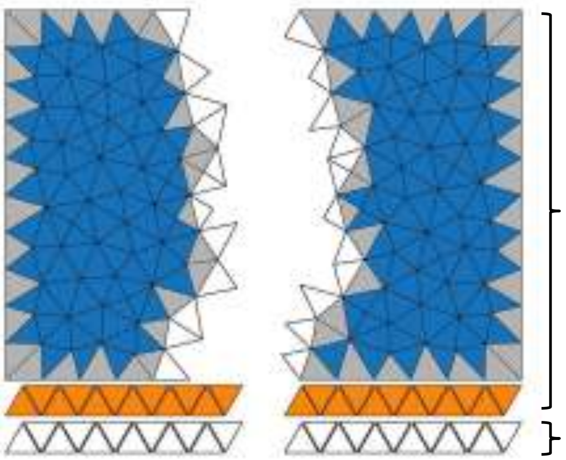




1. Launch the computation on the boundary
2. Pack the data that needs to be sent in a buffer
3. Copy to the CPU
4. Communicate with MPI
5. Copy back to the GPU

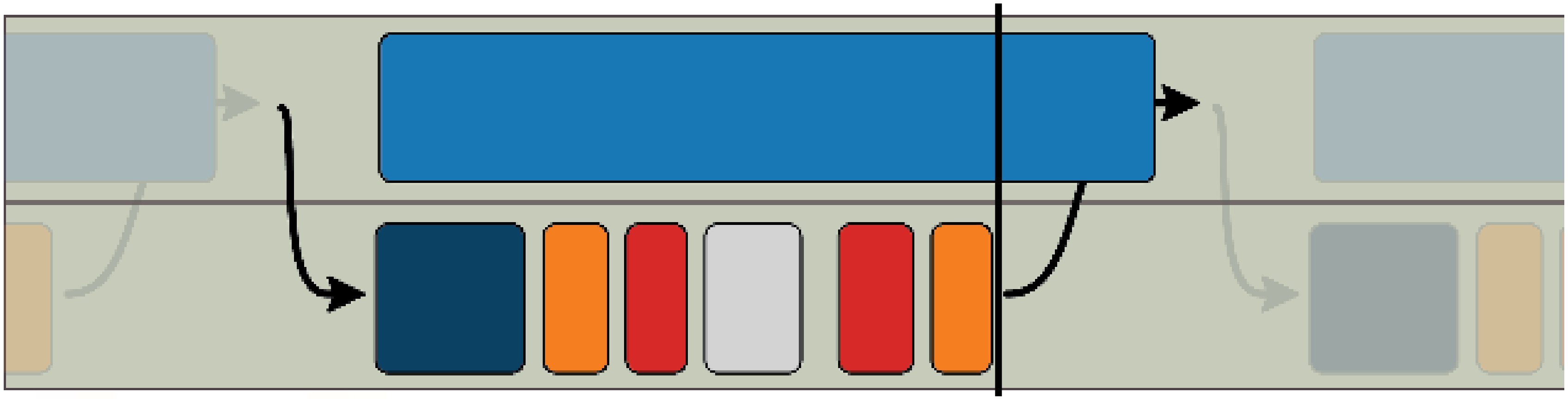
Structure of communications

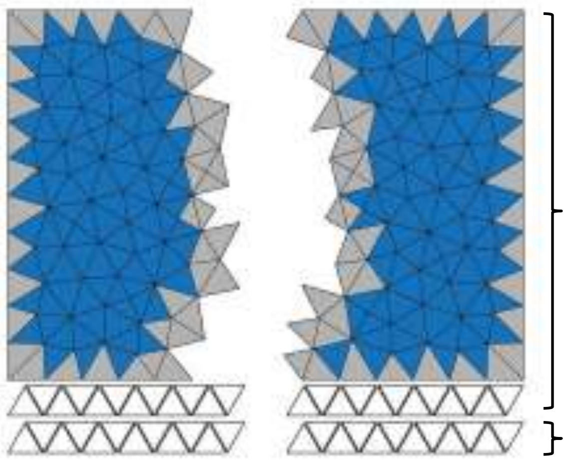




1. Launch the computation on the boundary
2. Pack the data that needs to be sent in a buffer
3. Copy to the CPU
4. Communicate with MPI
5. Copy back to the GPU
6. Unpack the arriving data

Structure of communications

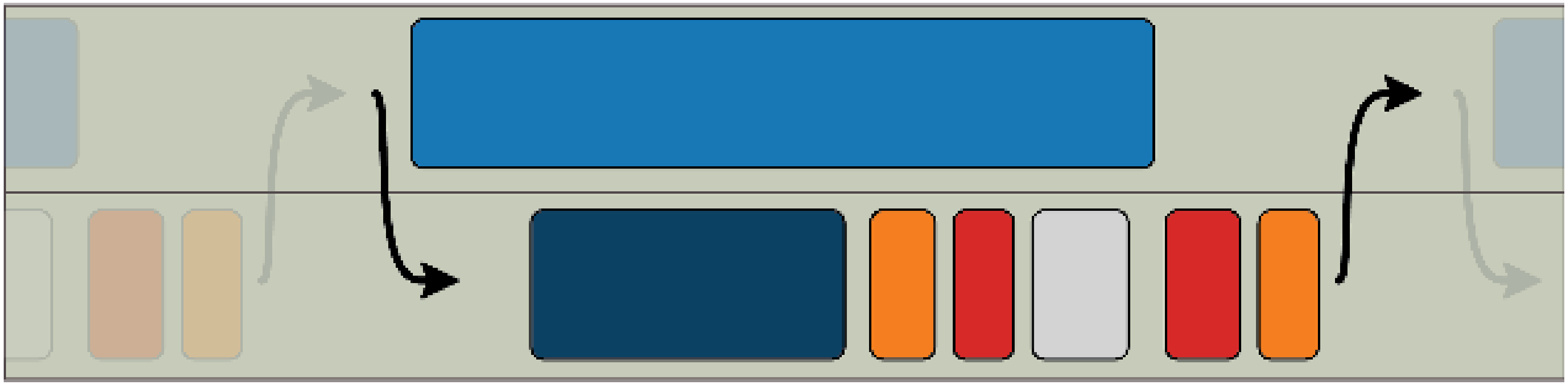


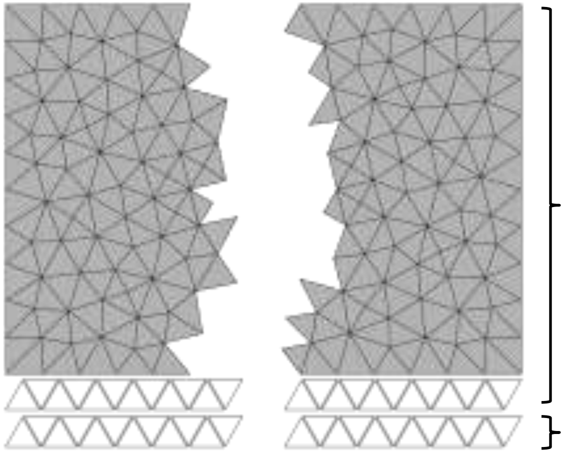


1. Launch the computation on the boundary
2. Pack the data that needs to be sent in a buffer
3. Copy to the CPU
4. Communicate with MPI
5. Copy back to the GPU
6. Unpack the arriving data
7. Wait for the computation to finish and synchronize

- Interior computations: specialized for no boundaries
- Works great for big kernels

Structure of communications

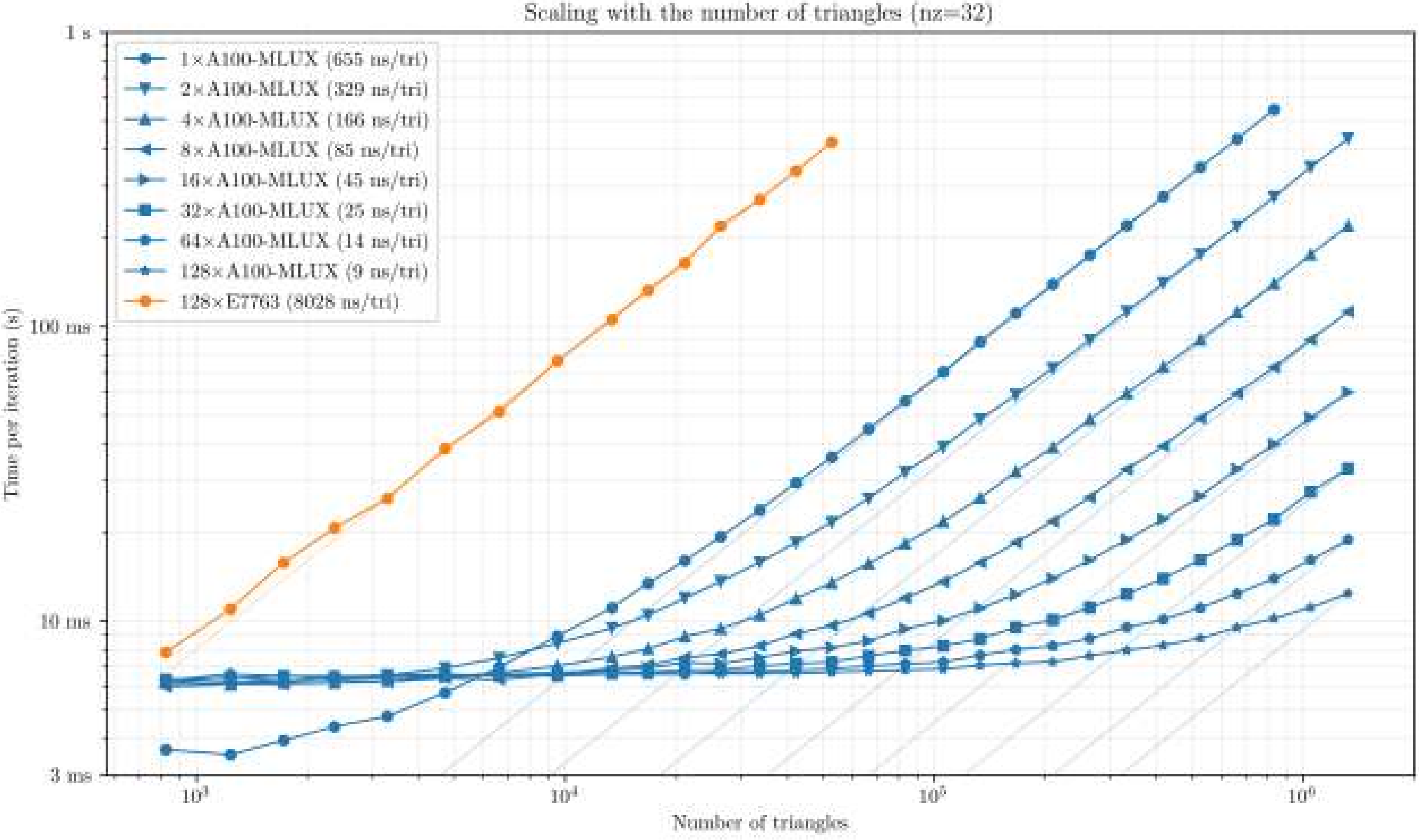




1. Launch the computation on the boundary
2. Pack the data that needs to be sent in a buffer
3. Copy to the CPU
4. Communicate with MPI
5. Copy back to the GPU
6. Unpack the arriving data
7. Wait for the computation to finish and synchronize

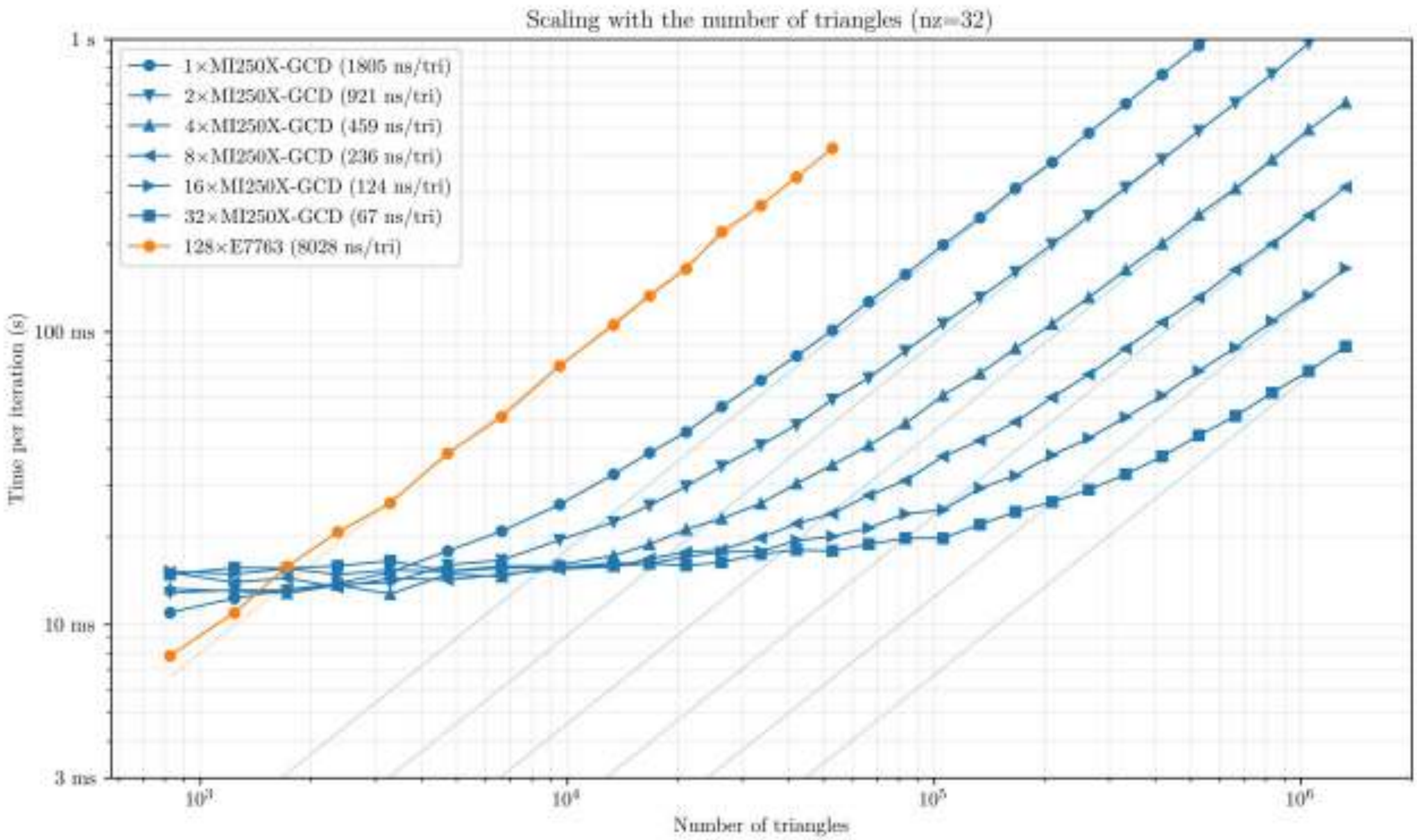
- Interior computations: specialized for no boundaries
- Works great for big kernels, but not so much for small kernels (<100µs)

Scaling of the 3D model on A100 GPUs



- Good single-GPU performance
- Good latency on single GPU
- Ok latency for multi-GPU

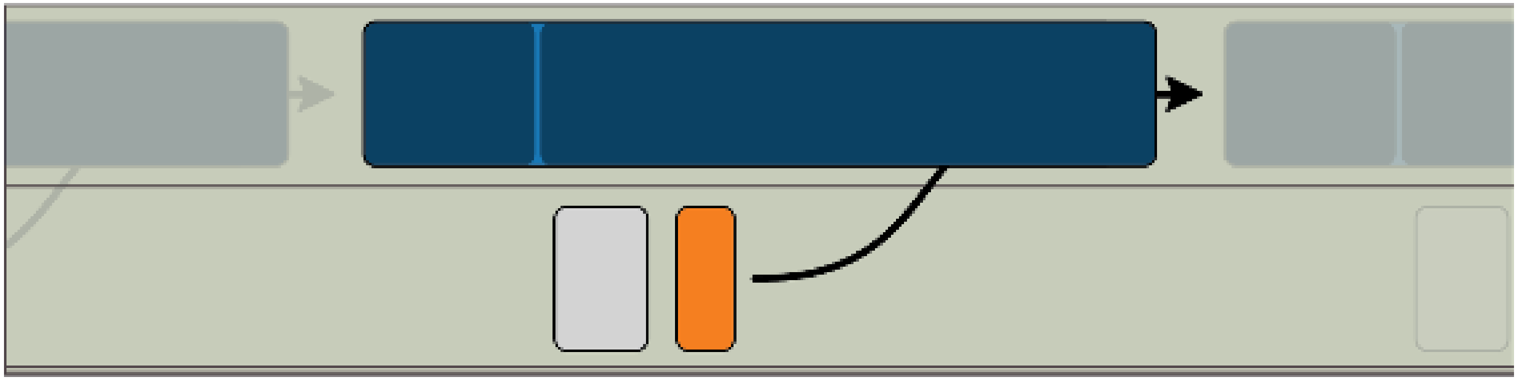
Scaling of the 3D model on MI250X GPUs



- Weaker single-GPU performance
- **Higher latency**
- OK-ish weak scaling

There are solutions to improve the MI250X perfs

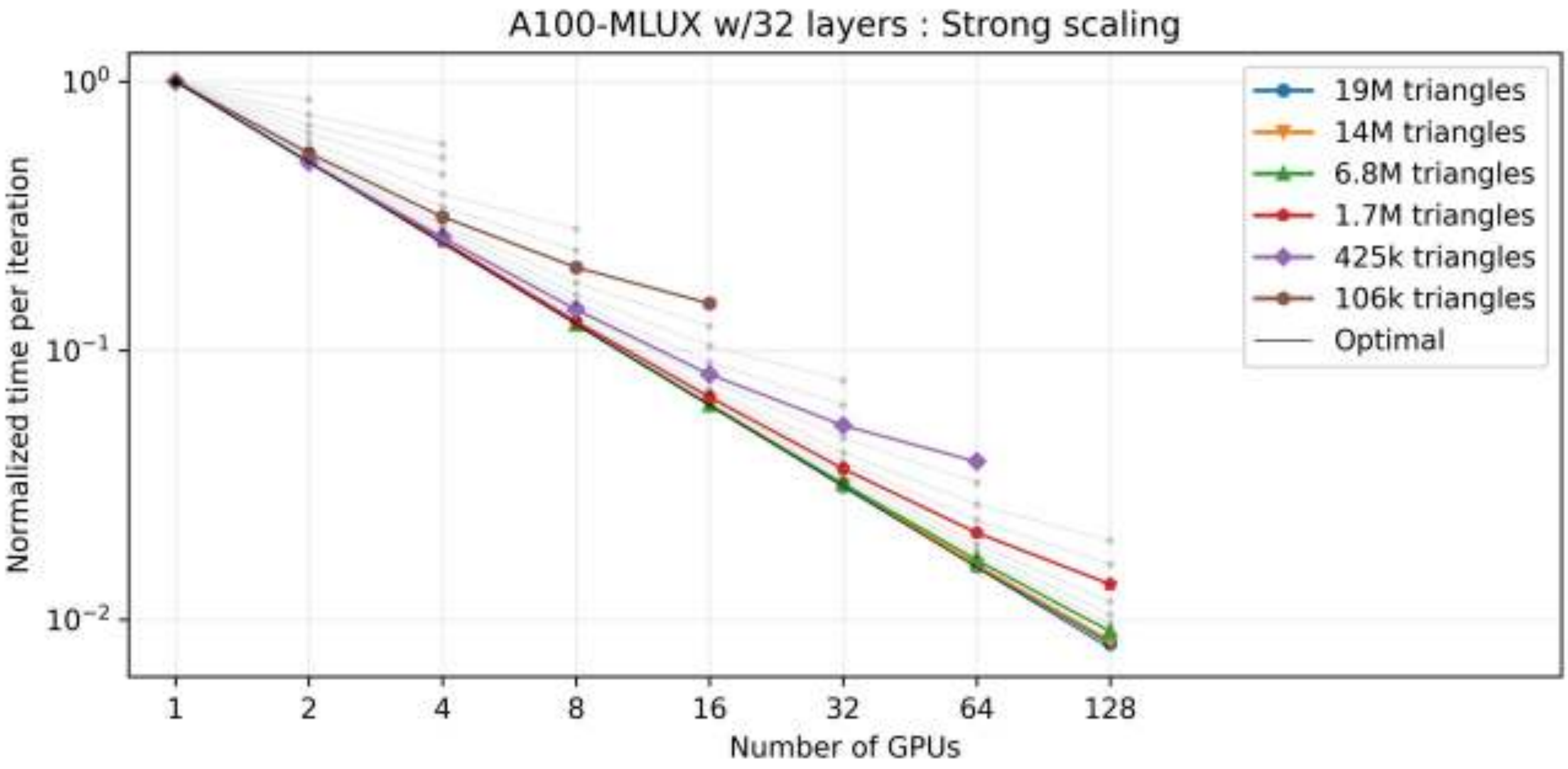
- GPU-aware MPI
- Compute on the boundary and pack the data within the same kernel
- Launch the full computation (inside and boundary nodes) and packing on the main kernel



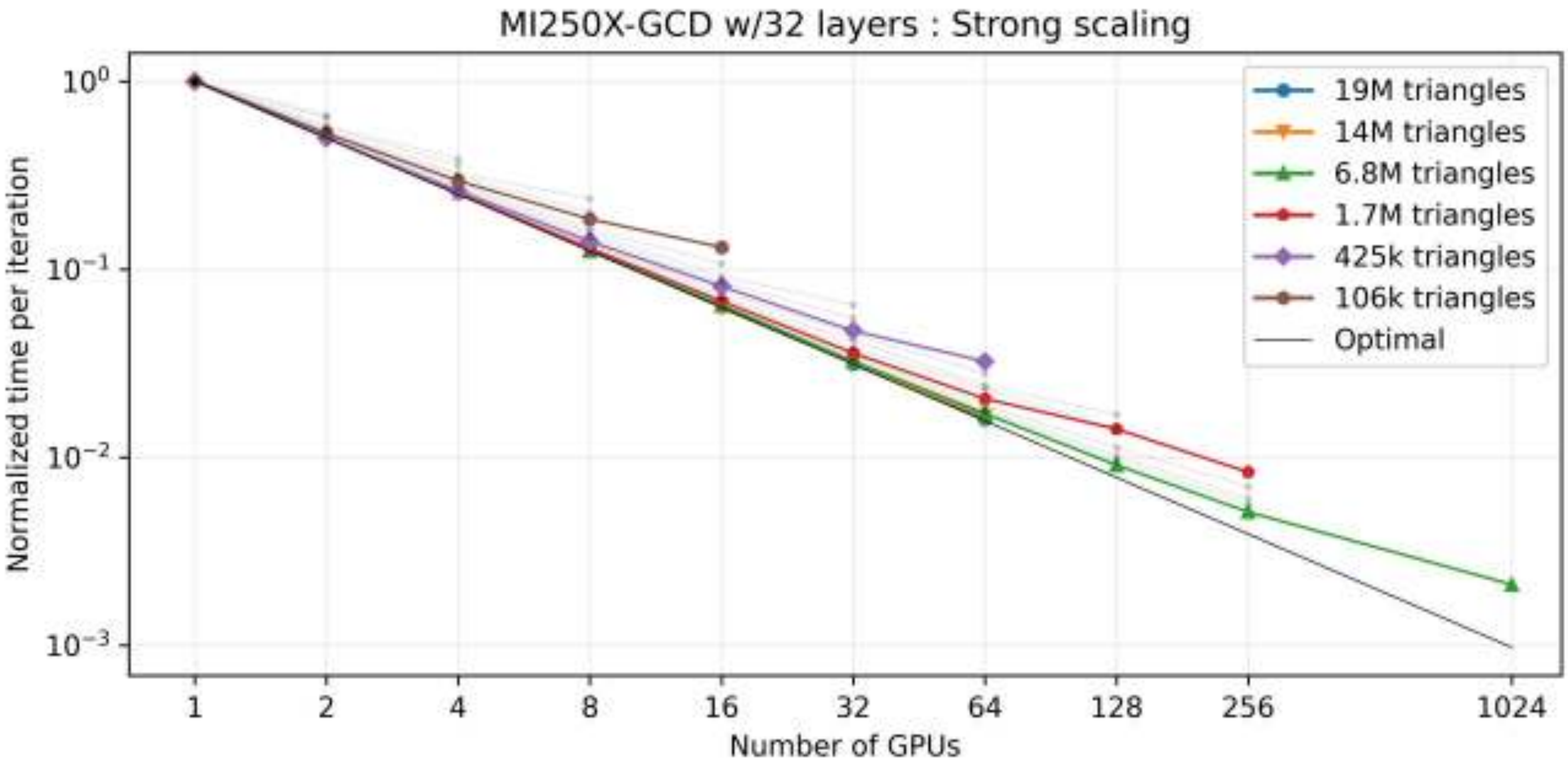
⇒ Leads to lower latency and better performances for AMD hardware

⇒ A bit slower on the NVIDIA hardware

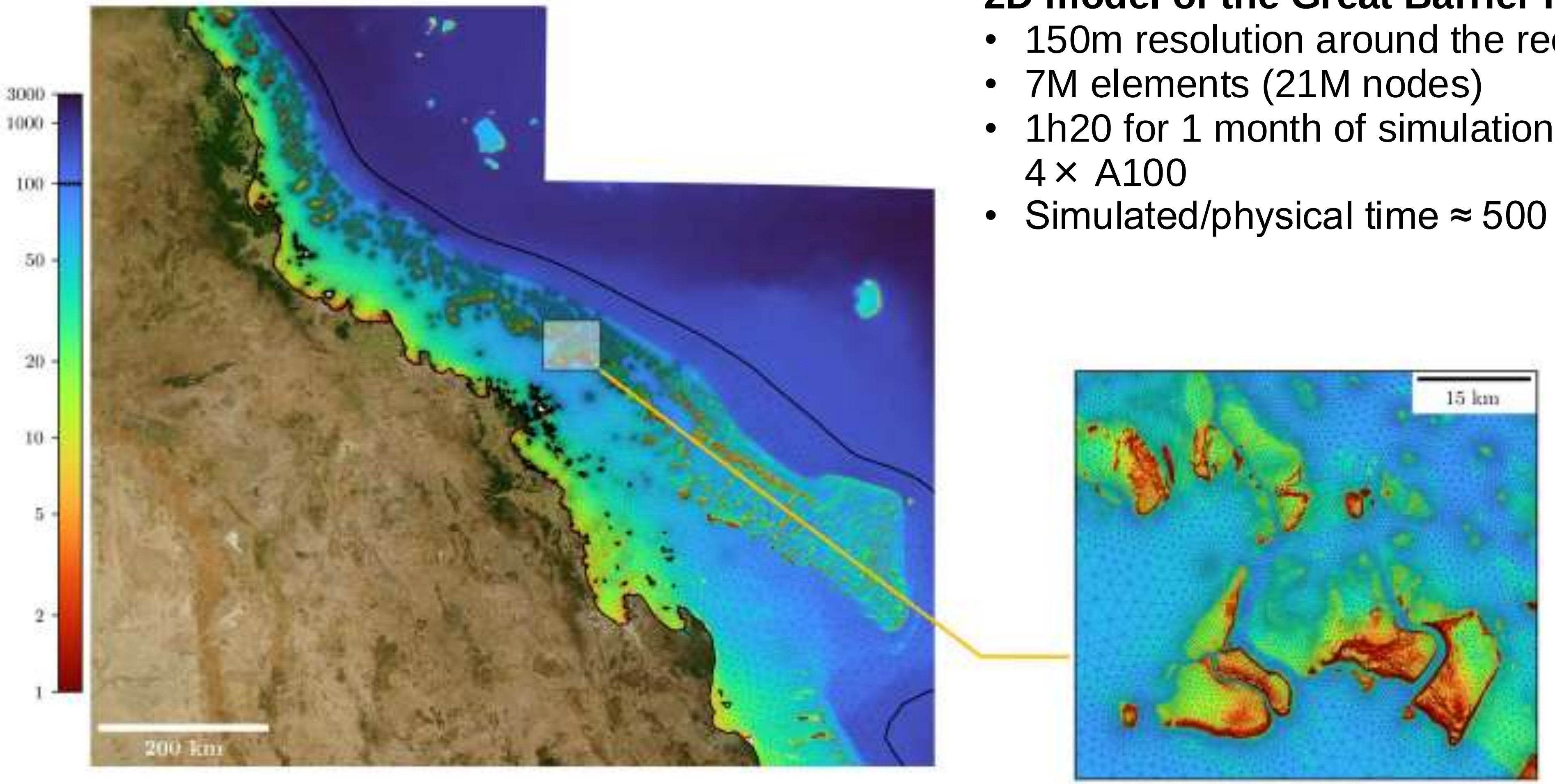
Strong scaling of the 3D model on A100 GPUs



Strong scaling of the 3D model on MI250X GPUs



And in the end, how well does it perform?



2D model of the Great Barrier Reef

- 150m resolution around the reef
- 7M elements (21M nodes)
- 1h20 for 1 month of simulation on 4 × A100
- Simulated/physical time ≈ 500

Conclusions

- Rewriting our code from scratch was a good investment!
- Simulations that used to take a few weeks now take a few hours.
- Achieving good strong scaling on 100's of GPUs opens the door to very ambitious simulations.
- Minimizing latency however remains an issue that requires hardware-specific solutions.
- The next step for us is to adapt the physics of the model to the new resolutions that we can now achieve…

Many thanks to LUMI-BE and EuroHPC for granting us access to those fast tools!