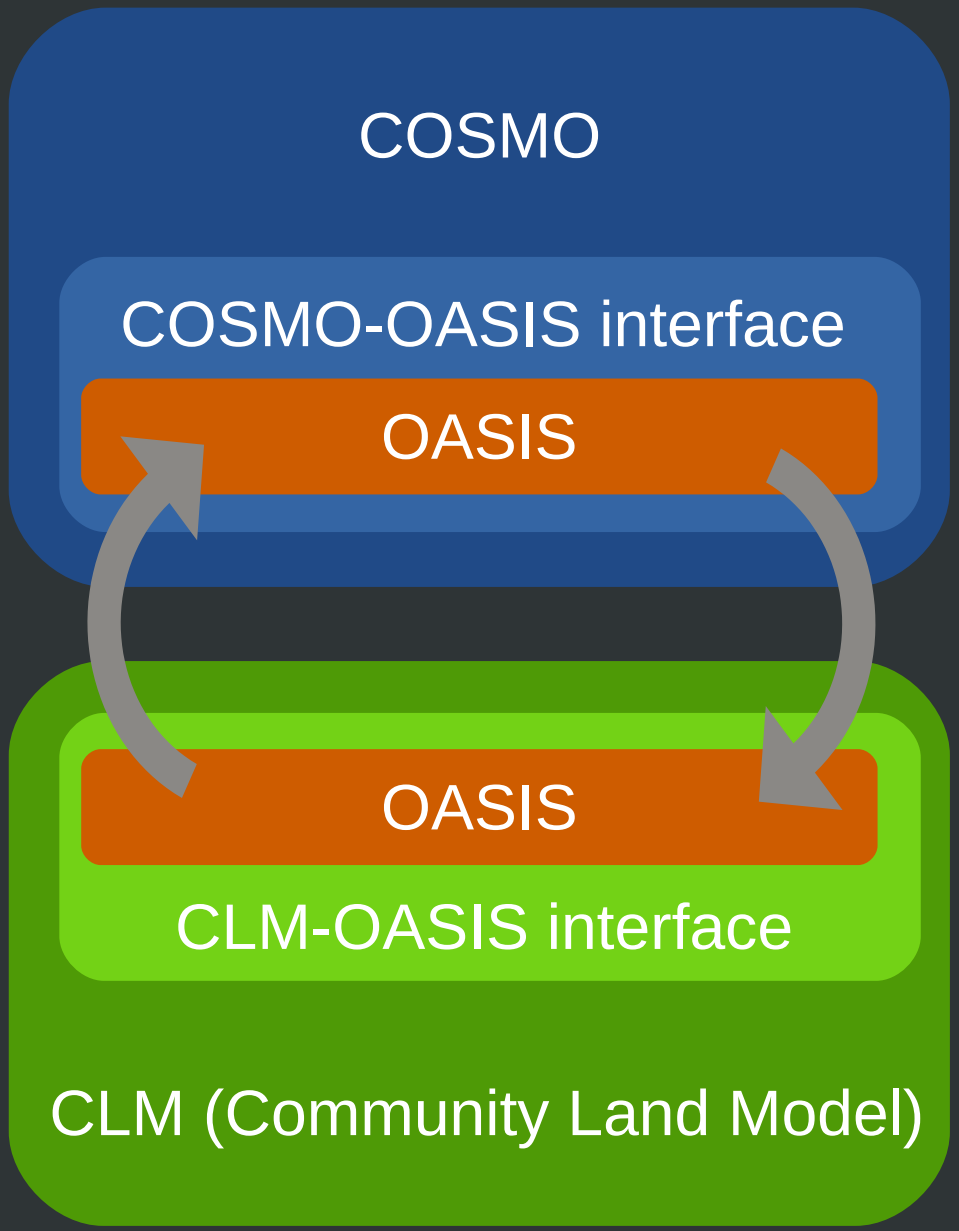
Porting COSMO_CLM² to GPU

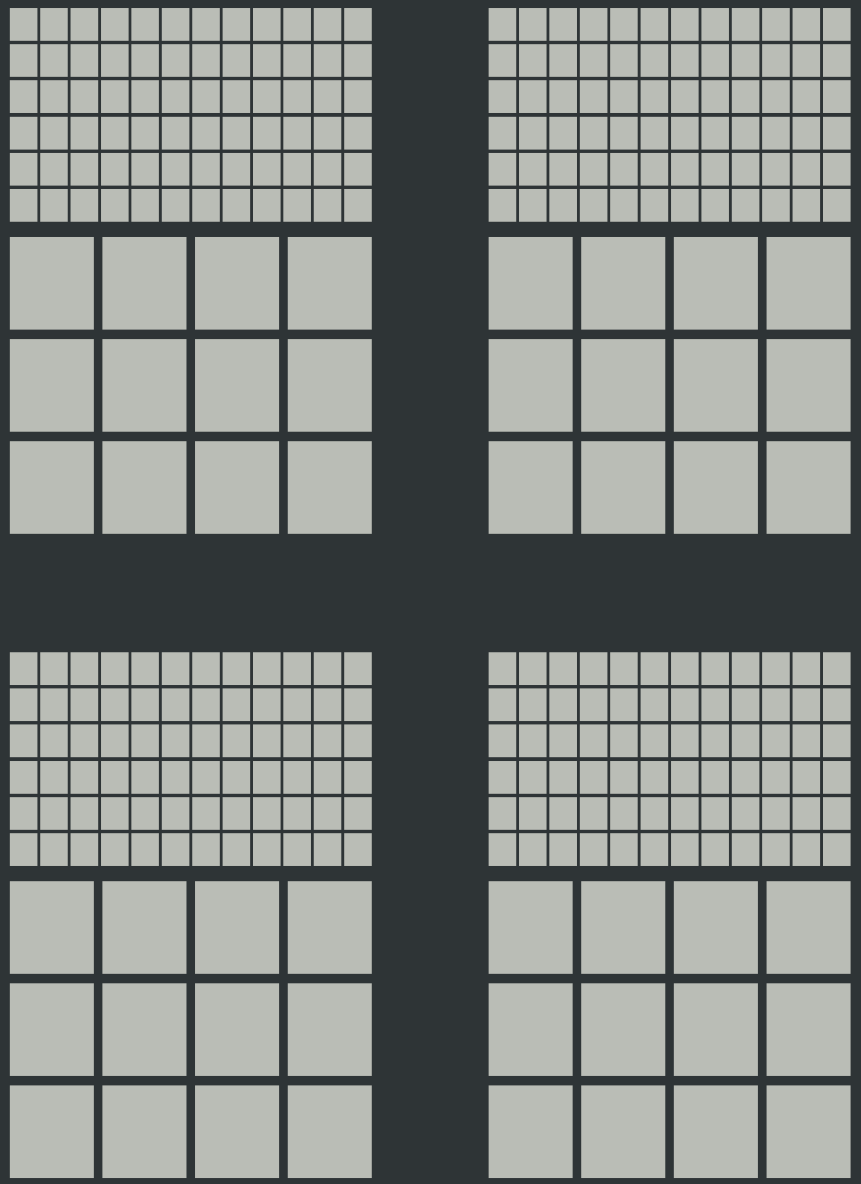
Matthieu Leclair & Edouard Davin

21th January 2019

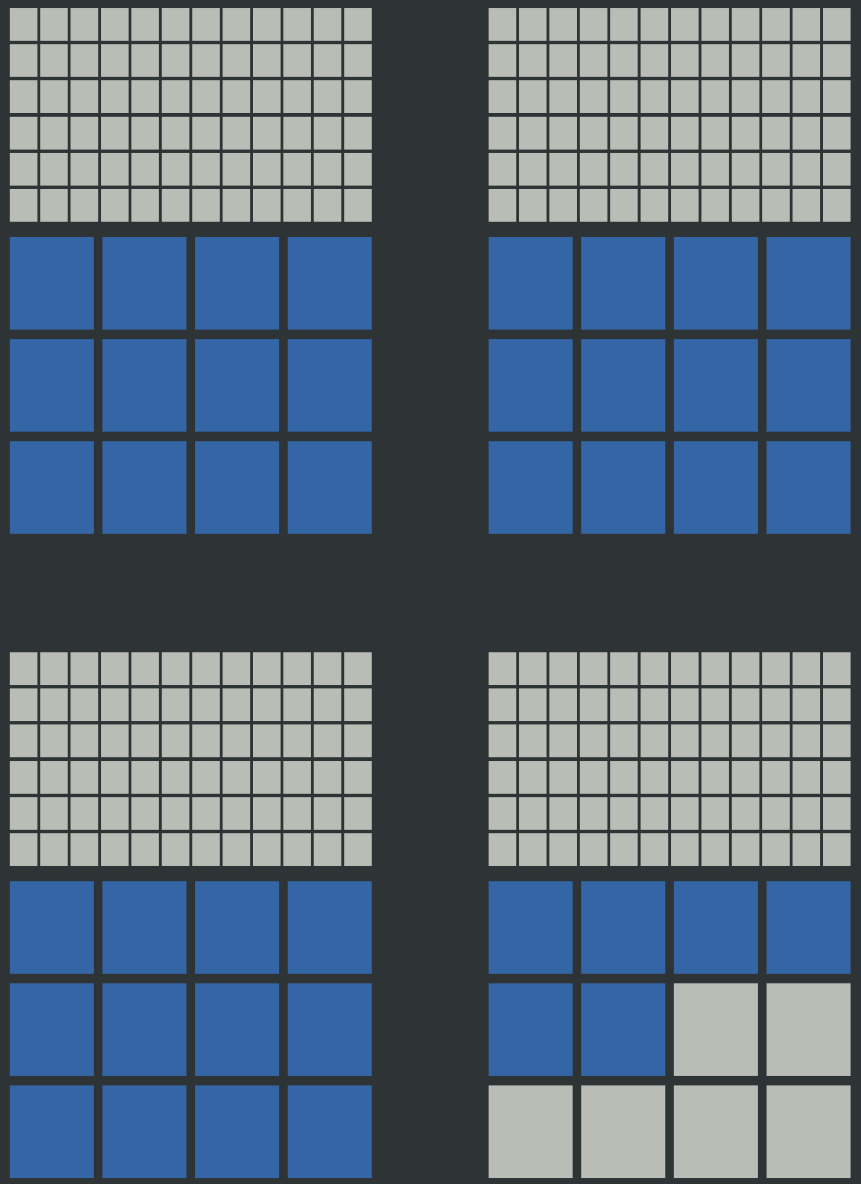
COSMO_CLM²


COSMO
COSMO-OASIS interface
OASIS
OASIS
CLM-OASIS interface
CLM (Community Land Model)

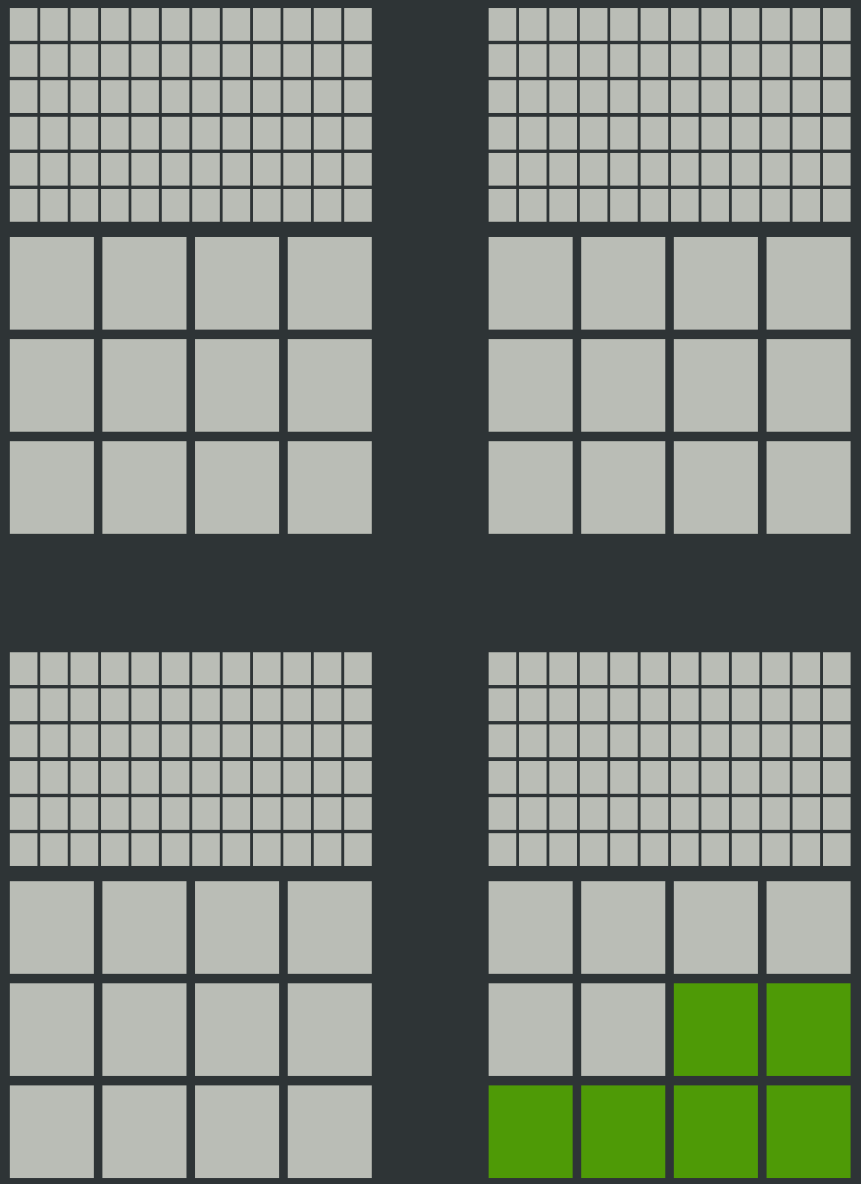
Daint nodes



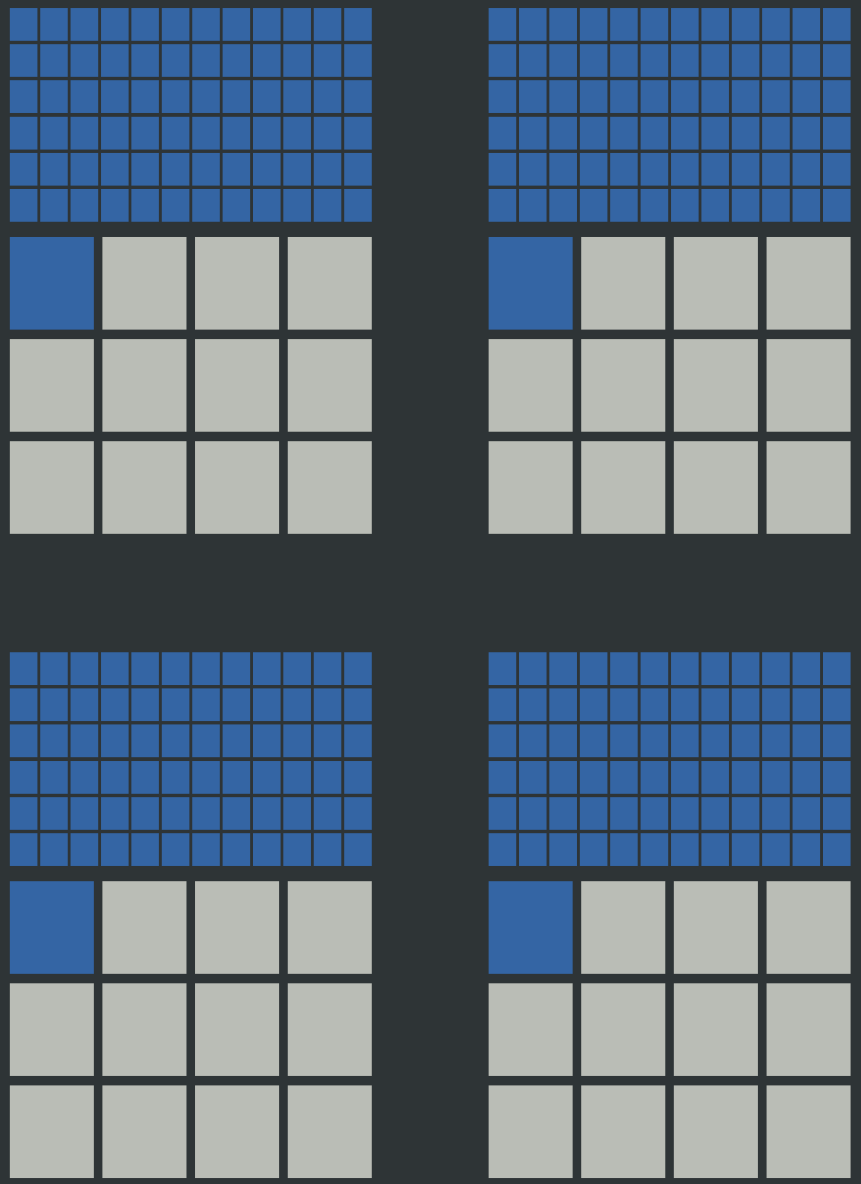
CPU implementation – COSMO phase



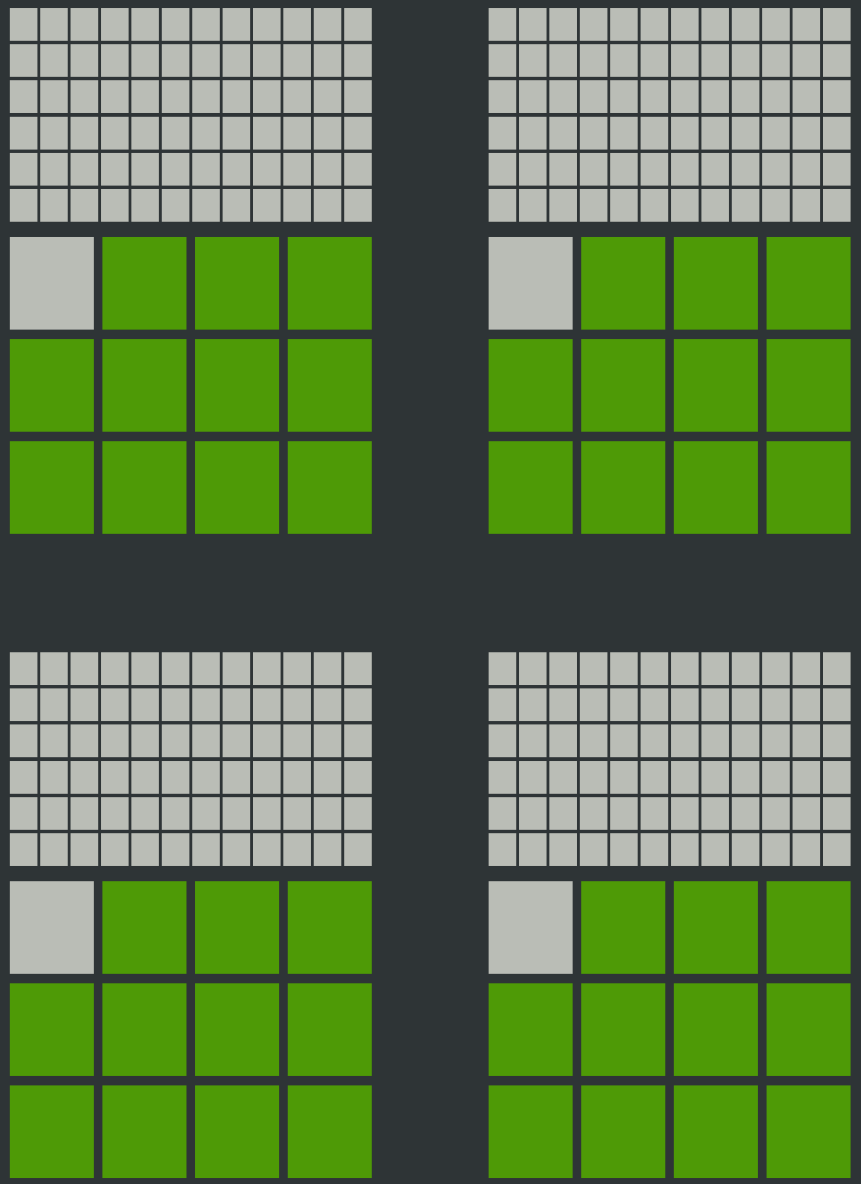
CPU implementation – CLM phase



GPU implementation – COSMO phase



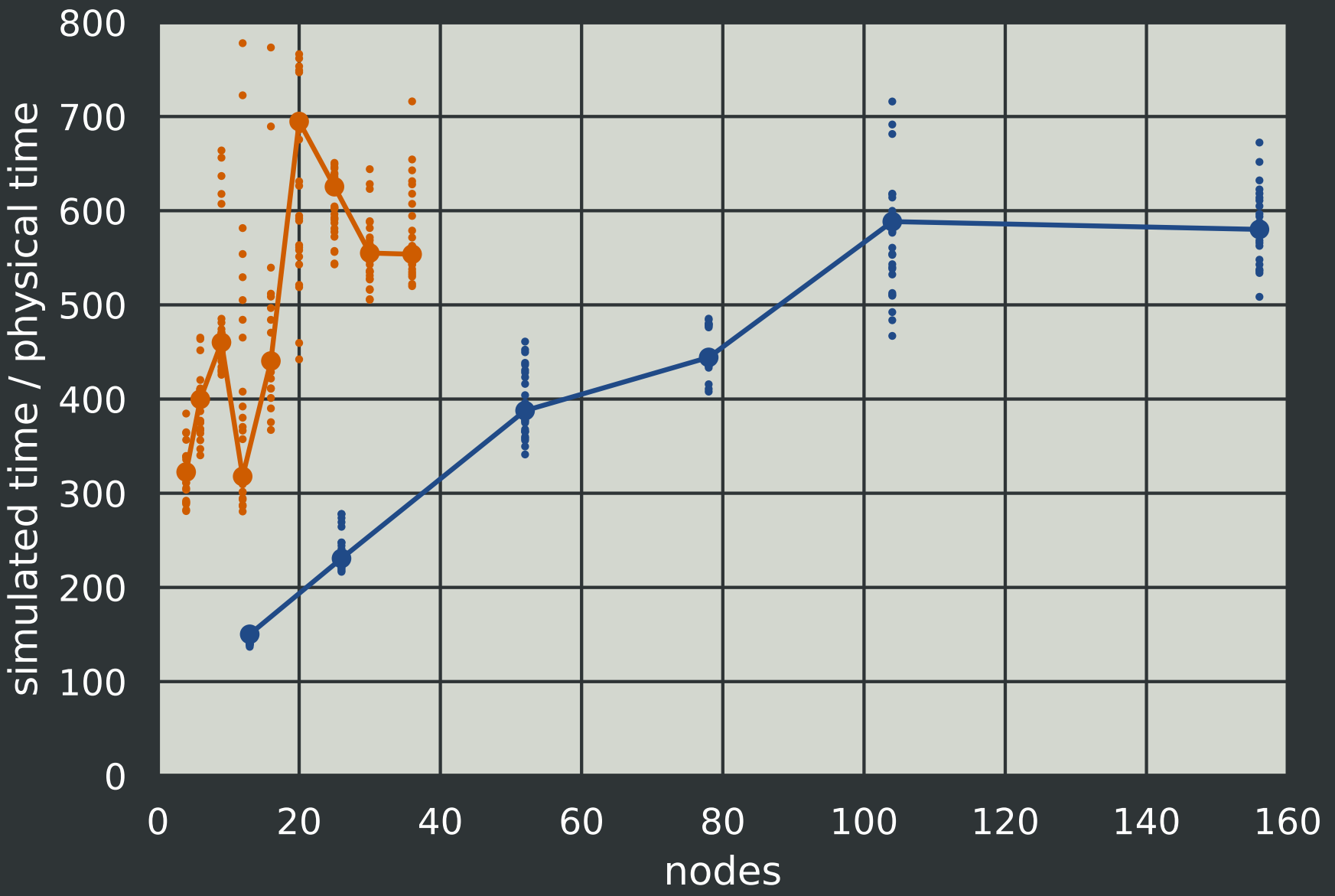
GPU implementation – CLM phase



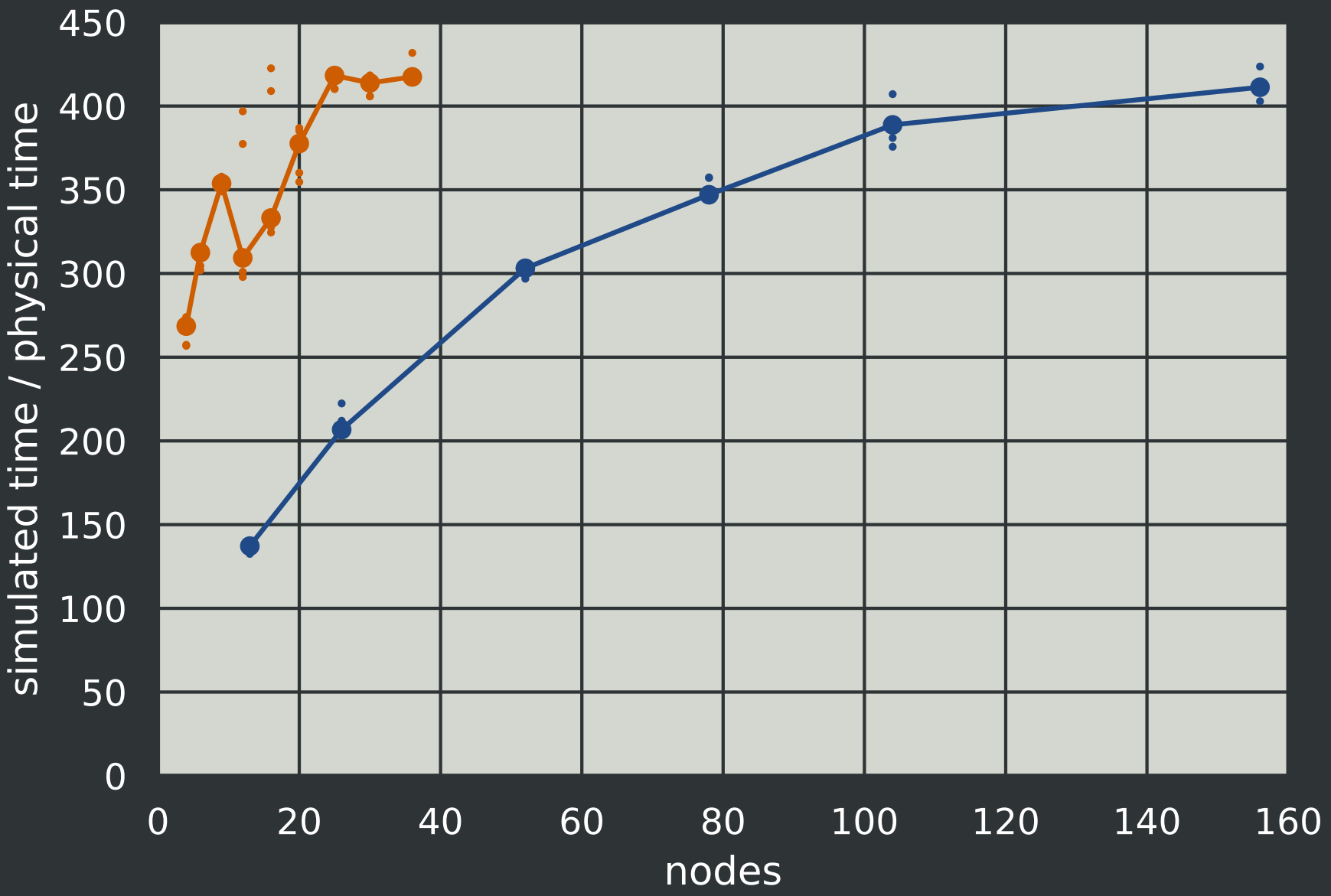
Performances – Setup

- COSMO-7 domain 393 x 338 x 60
- Not ideal for decomposition: $393 = 3x131$ and $338 = 2x13x13$
- Simulations of 2 days and 1 day, then difference => remove initial and final non scalable parts
- A lot of variability!

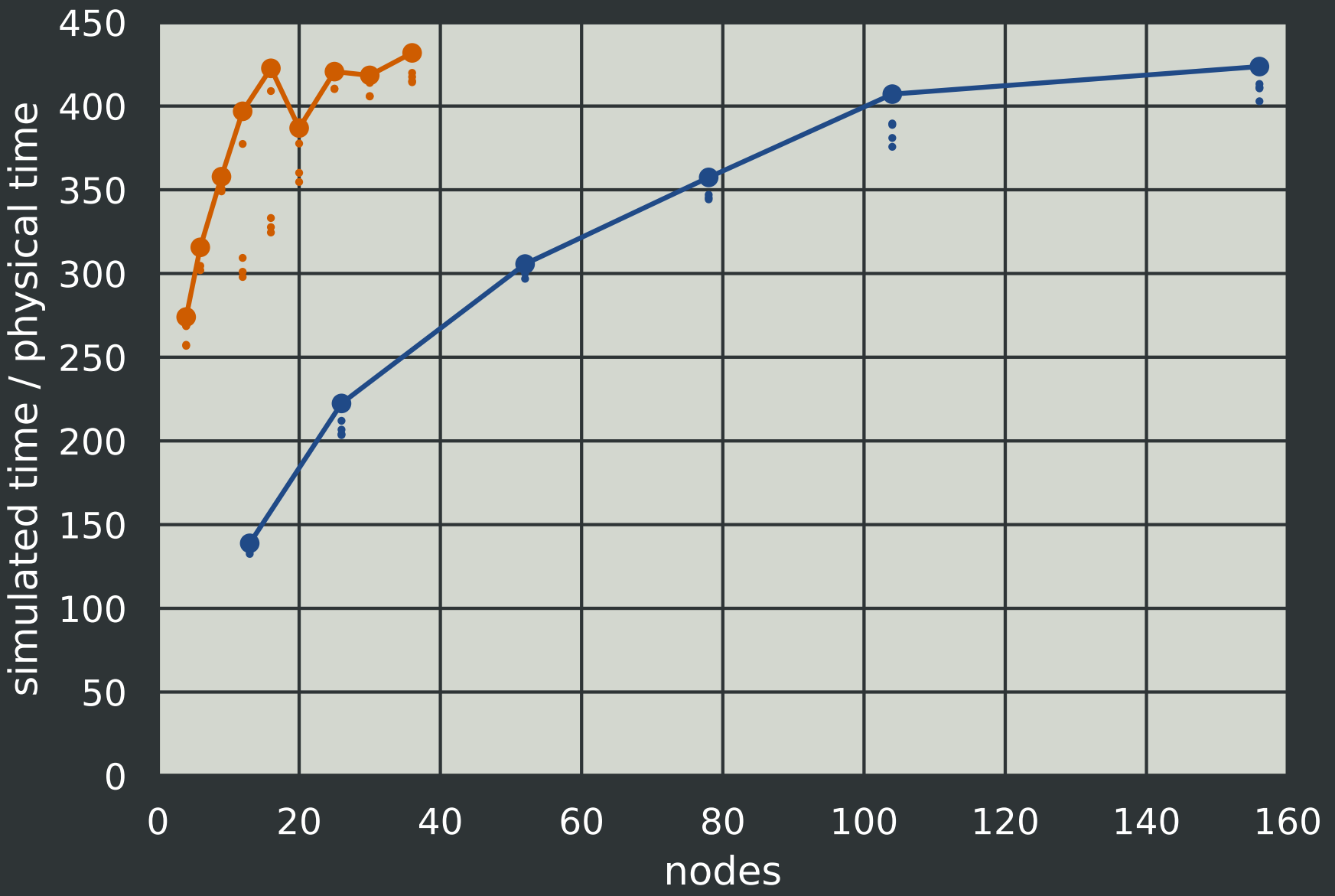
Performances – median per simulated day



Performances – median for 2 days simulations



Performances – best for 2 days simulations



Momentum coupling

Coupling from CLM to COSMO is done through boundary condition fluxes in vertical diffusion

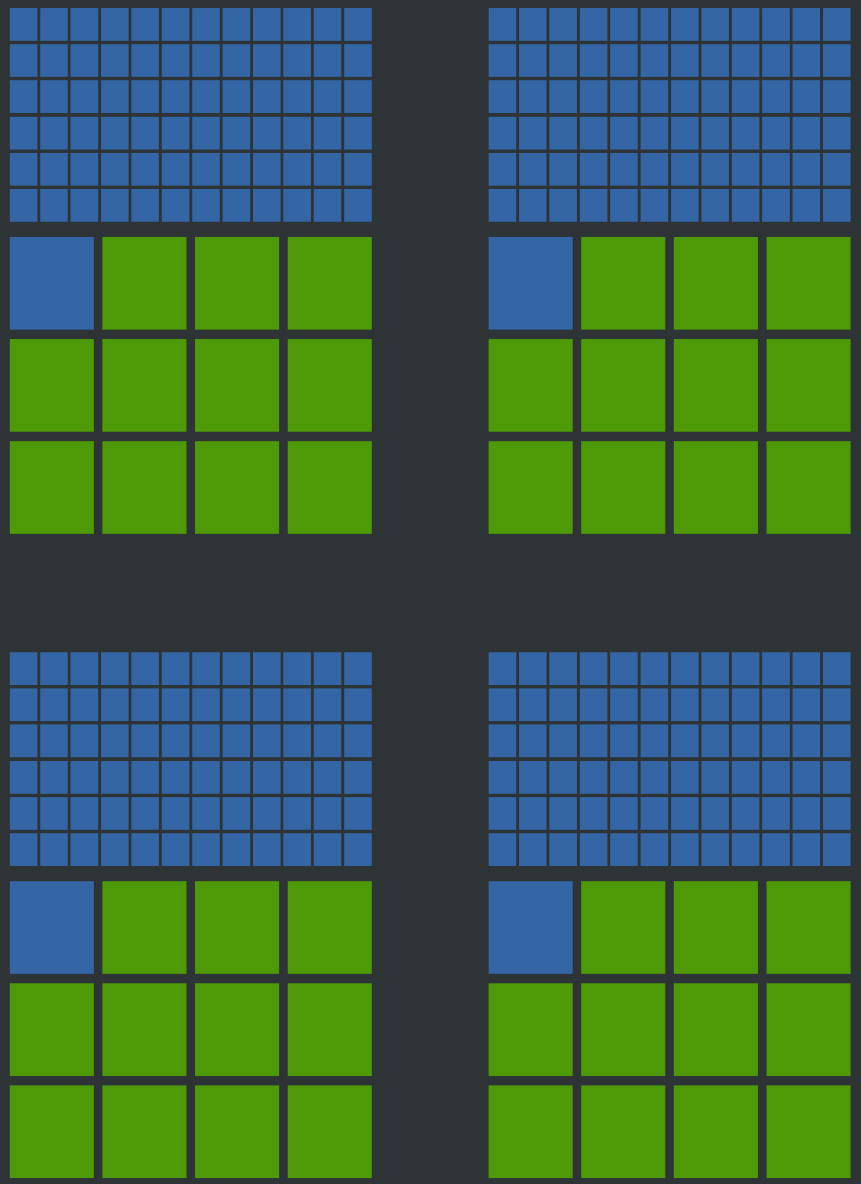
Is that right ?

- Heat, humidity -> fine!
- Momentum, not so sure . . . Rather mixing coefficient (surface roughness)

Help !

- vertical diffusion inversion
- dycore

Asynchronous coupling



Other optimizations

- Take advantage of COSMO single precision possibility. Need some help on the OASIS side.
- Port some of OASIS code to GPU

COSMO_CLM² tools

Python command line tool for simulation management

- Ensure coherence between COSMO and CLM namelists
- Clear and trackable cases set up through xml file and/or command line arguments
- Easy sensitivity experiments
- Works also with COSMO only
- Used at DKRZ
- ...

Thanks!