Task 7.2e: EC-EARTH 3 progress report

~

CAPACITIES

e-infrastructure

Independent scaling analysis using scalasca

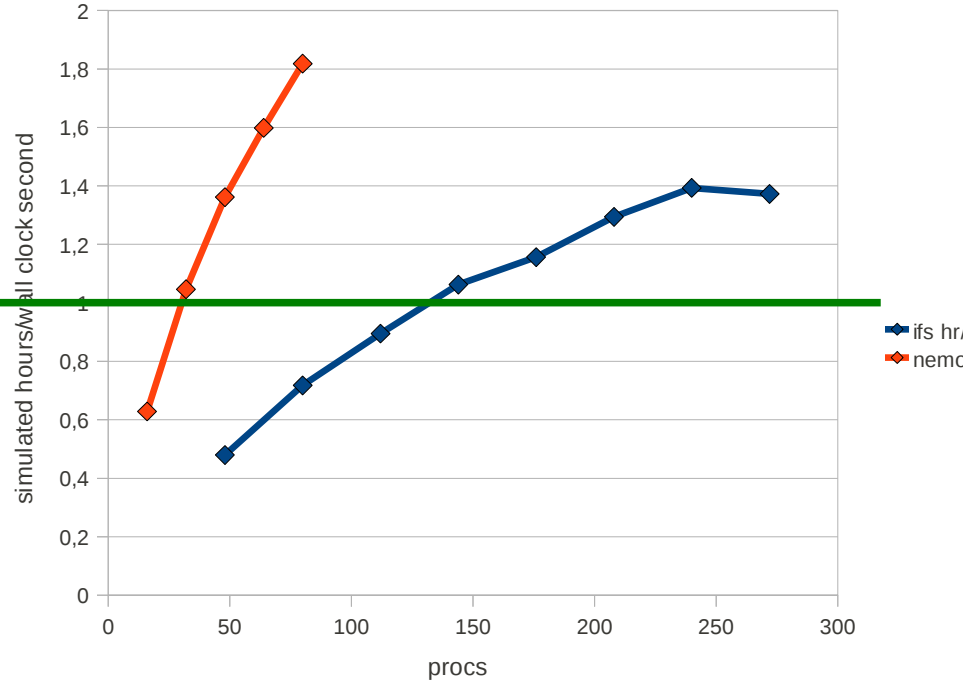
Determine total time on all cores for the routines:

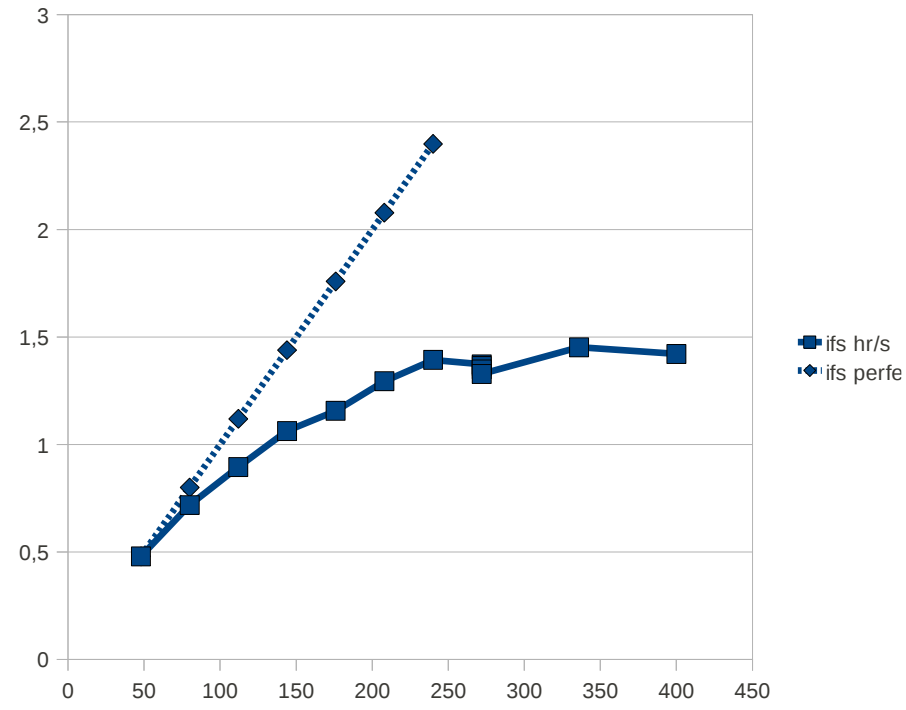
- cnt0 – total time for IFS main loop
- stp – total time for NEMO main loop
- prism_{put,get}_proto_r18 – coupling routines in IFS
- cpl_prism_{snd,rcv} – coupling routines in NEMO
- sbc_cpl_init – initialize coupling in IFS
- setoasis3 – initialize coupling in NEMO

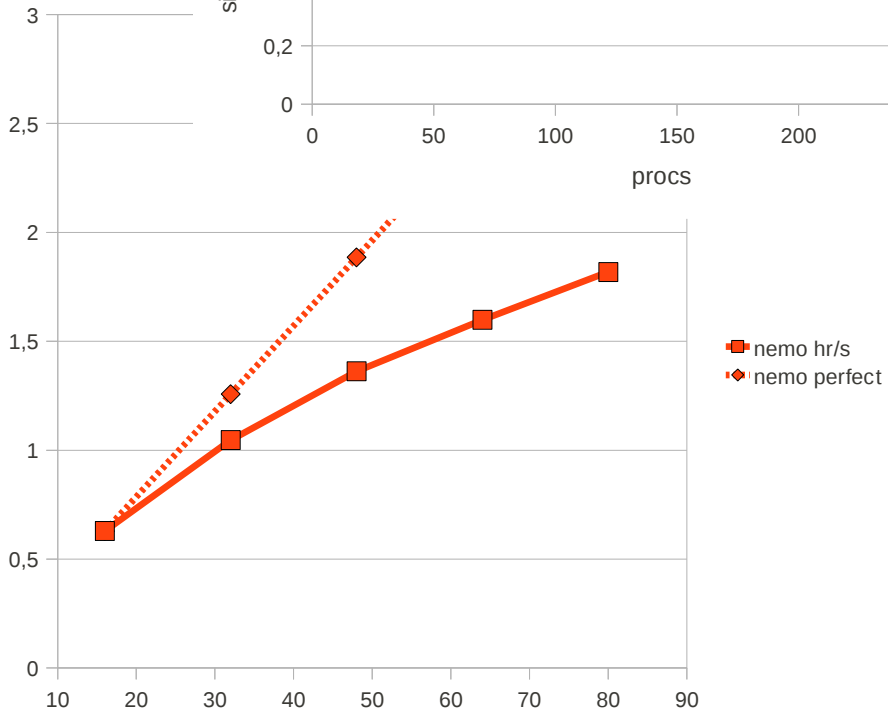
time for model component = total – coupling - init

Scaling on huygens

10 yrs/day

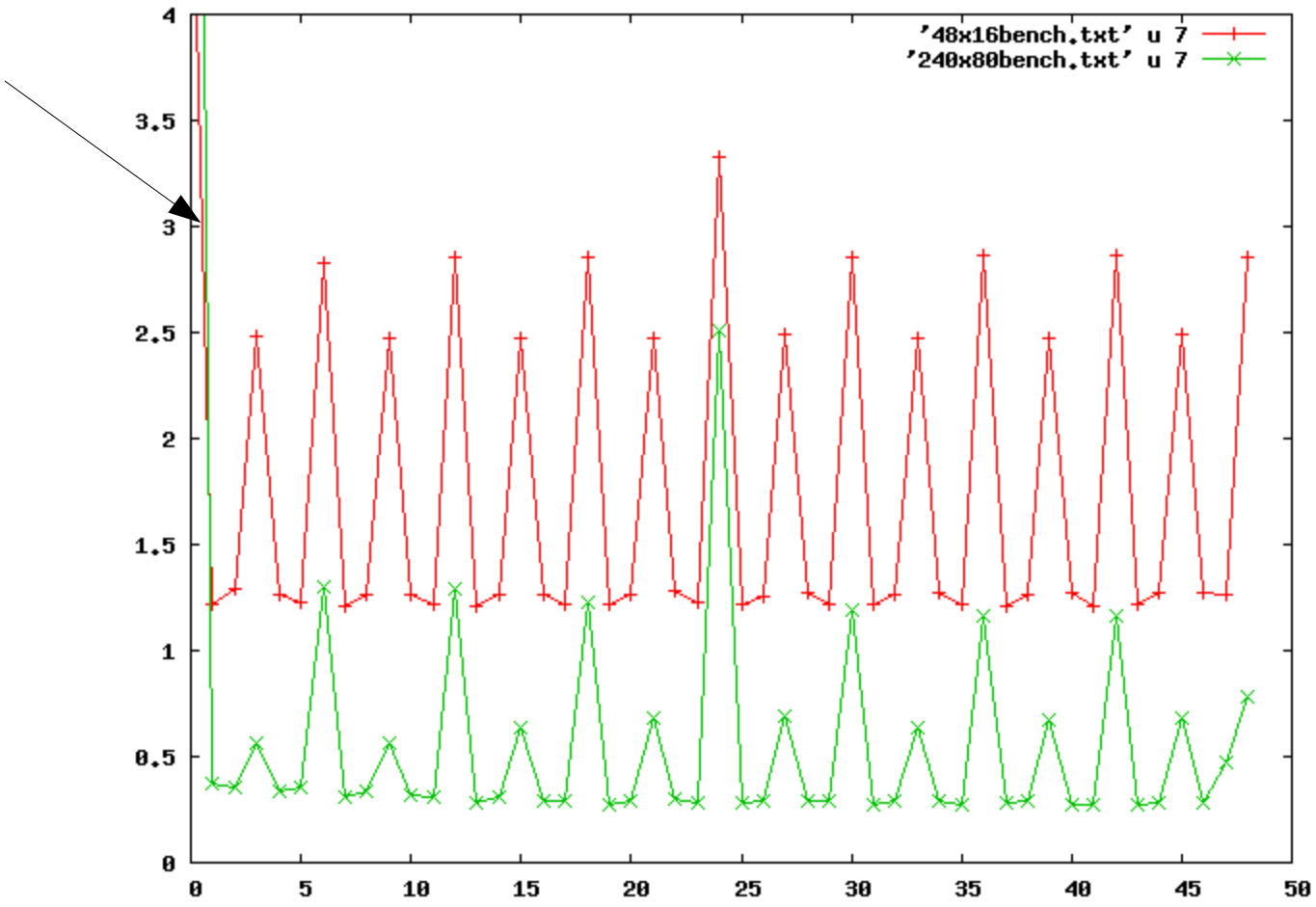




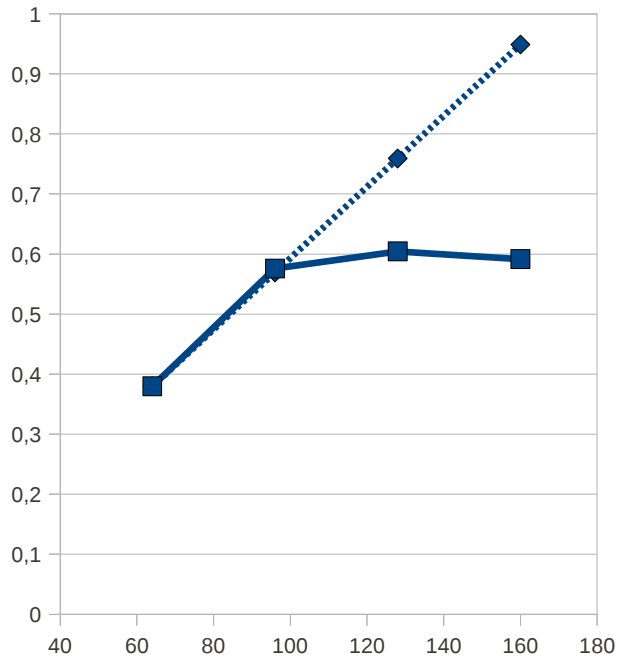


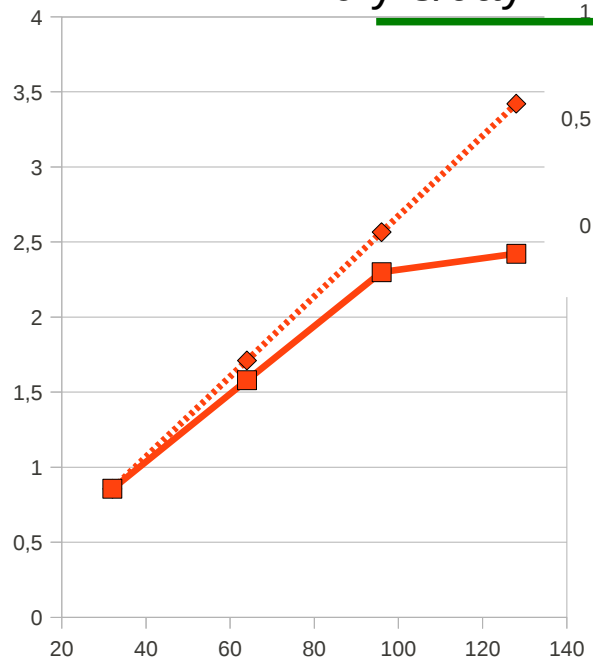
Scaling on huygens: timesteps

init

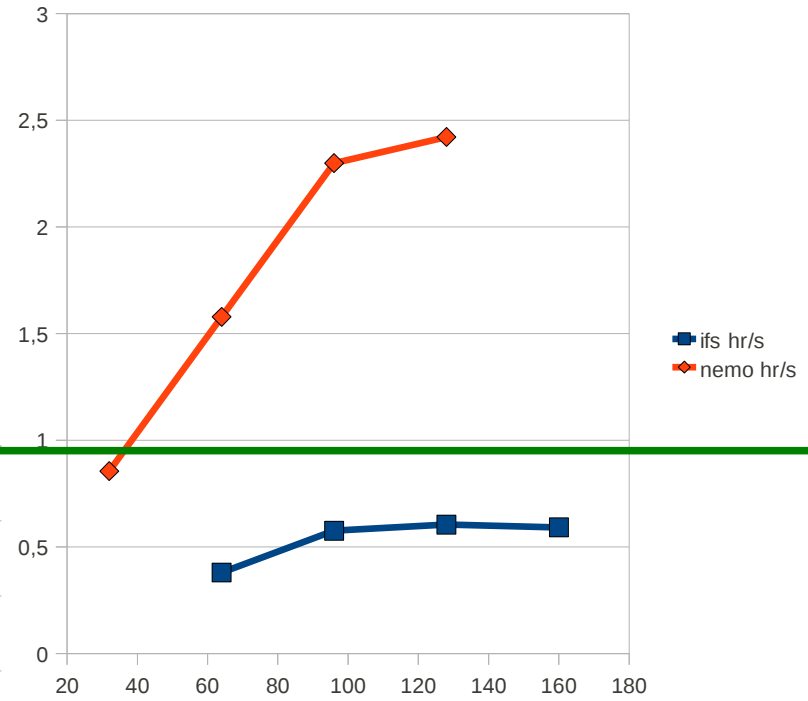


Scaling on Curie





10 yrs/day



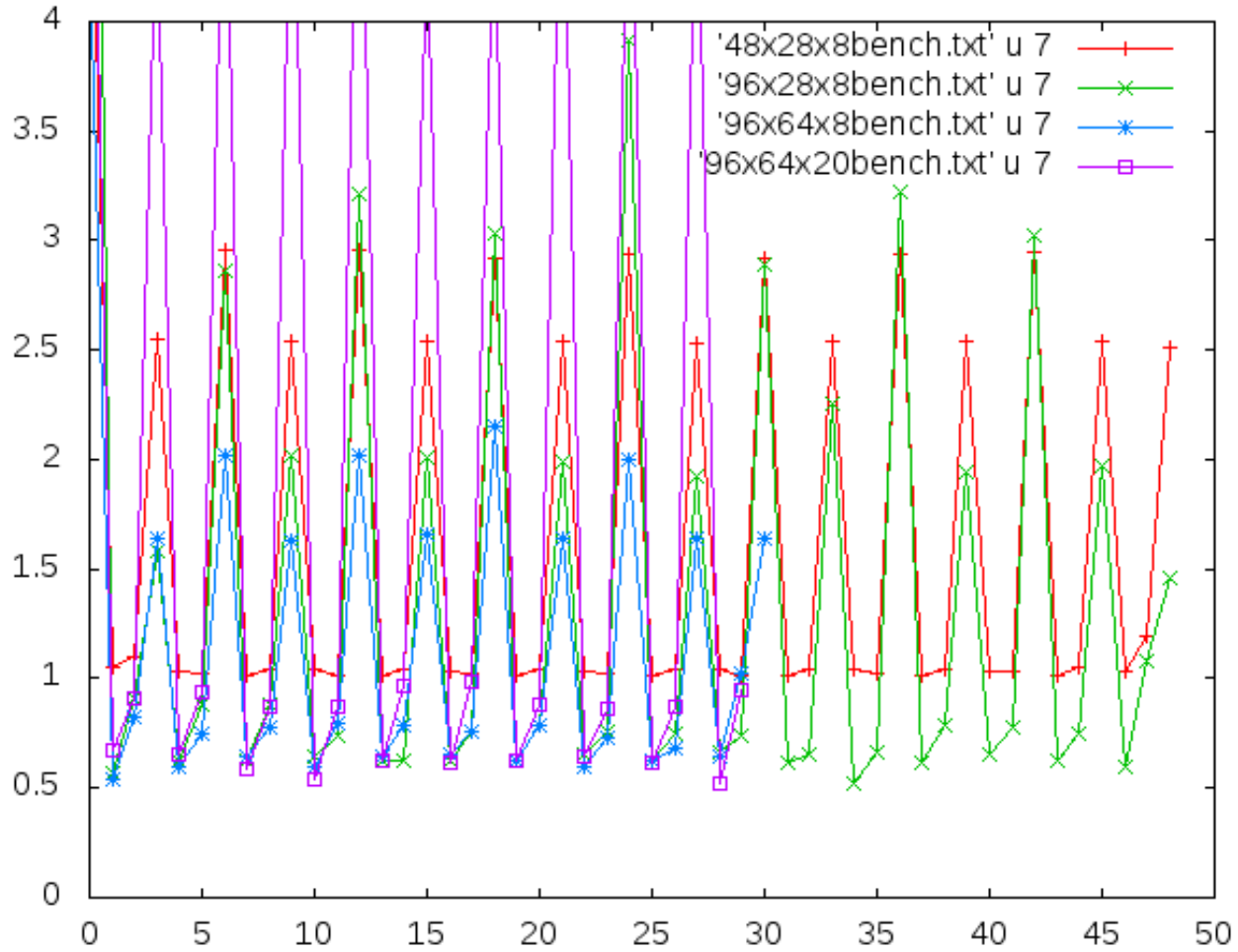
Scaling on Curie: timesteps

nr. of NEMO tasks chosen
to be faster than IFS model

medium: 4

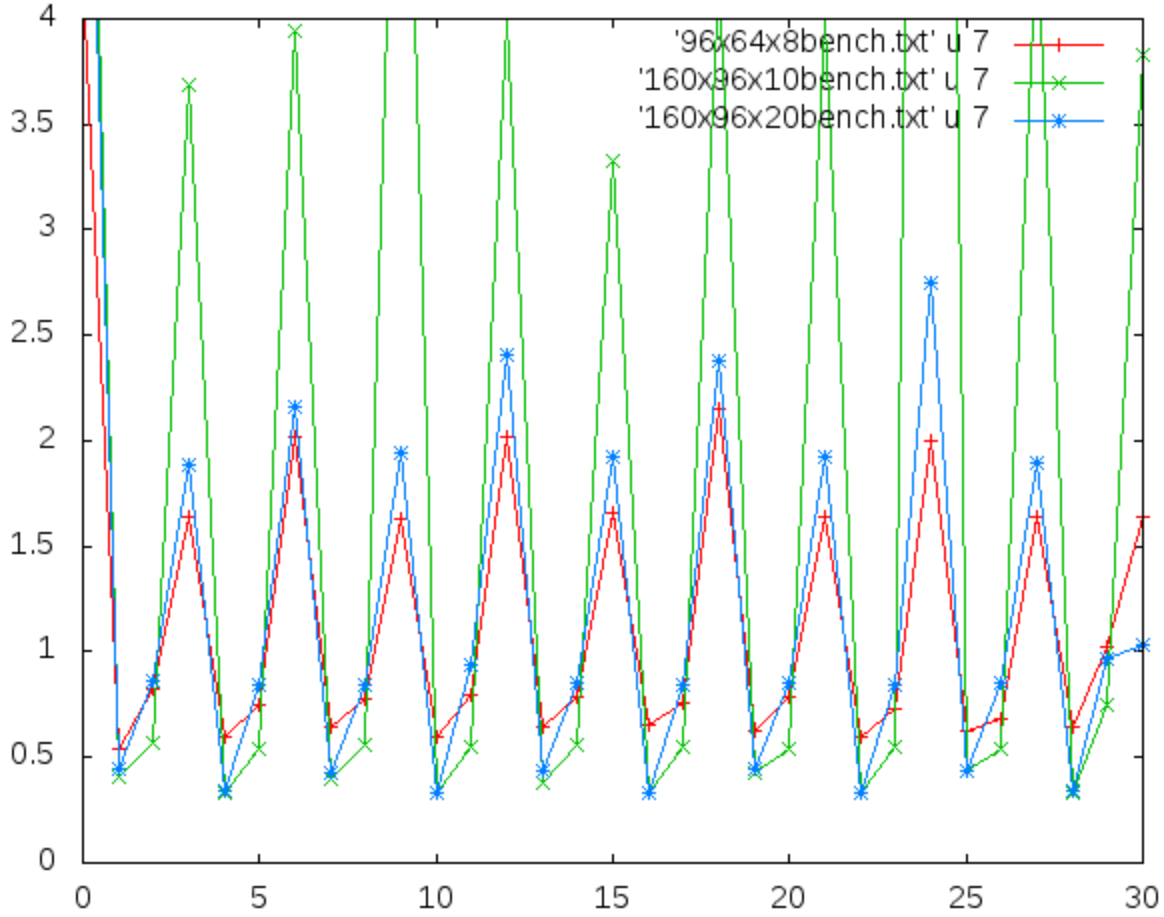
long: 83% loss

short: 30%



NEMO needs
to be significantly
faster than IFS!

Scaling on Curie



It seems beneficial to run at scale with as many OASIS processes as coupling fields.

OpenMP on Curie

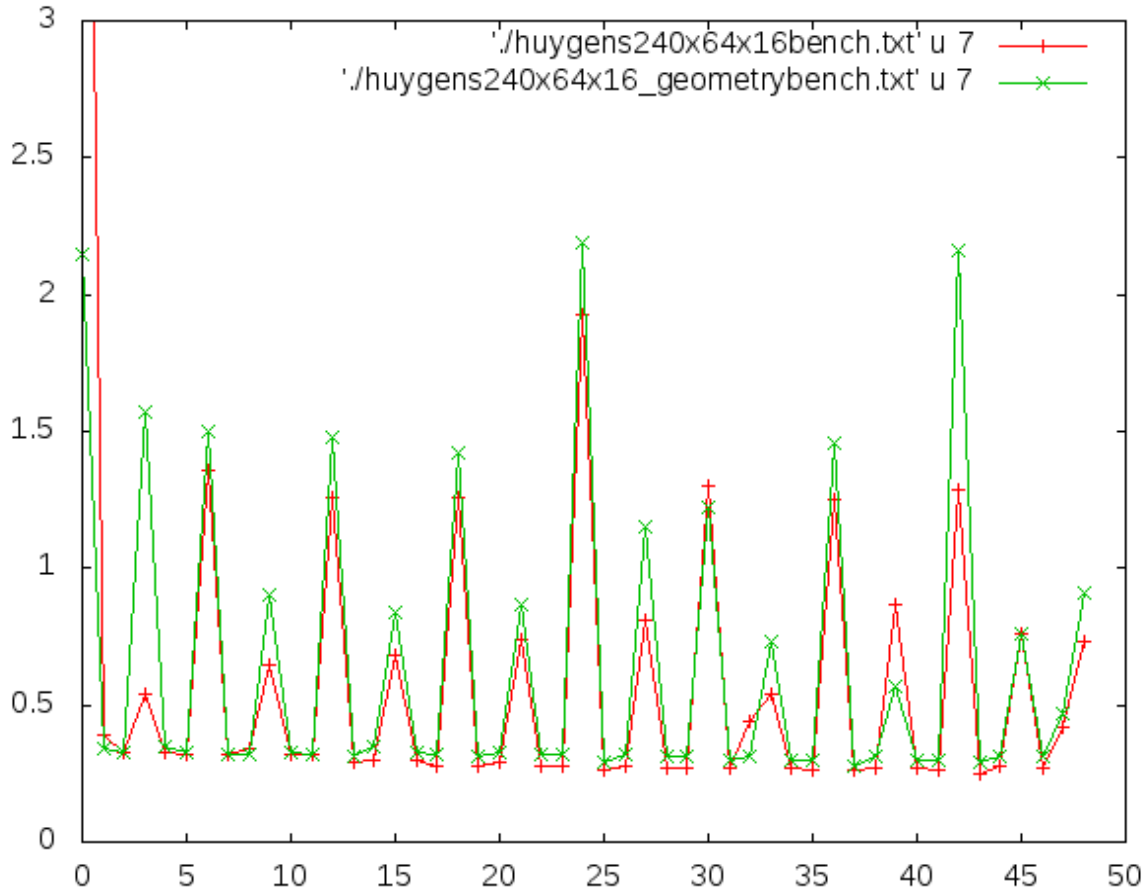
- Goal: Use OpenMP to lower communication time between MPI tasks.
- Straightforward to compile and run with OpenMP on Curie:
  - use OMP_STACKSIZE=1G
  - set affinity for each thread
  - use active wait policy
- Unfortunately, OpenMP does not give any performance improvement on Curie.
- OpenMP on Huygens (Power6) crashes.

Mapping of processes and threads

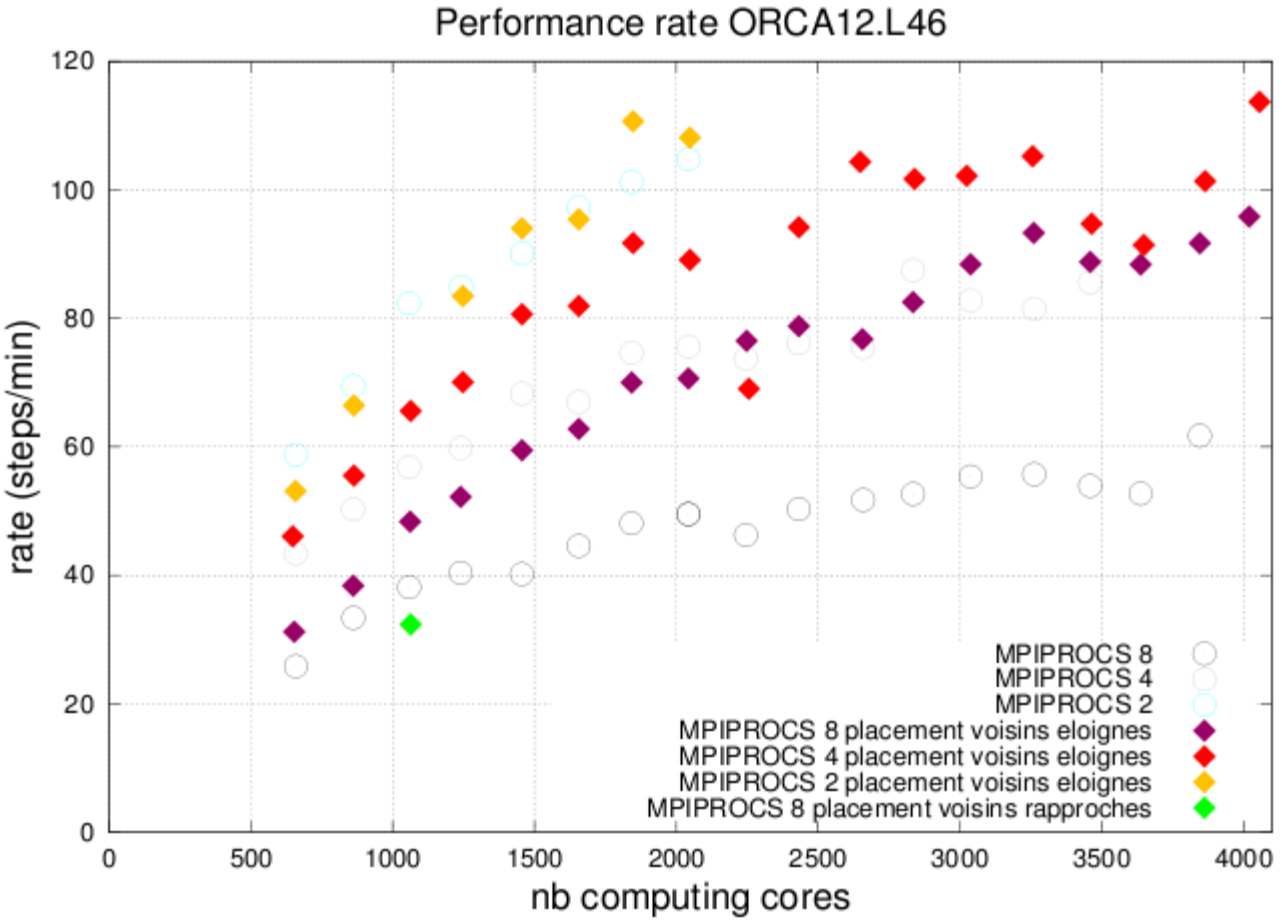
Goal: prevent coupling communication overload on first node, distribute OASIS processes across nodes.

- OpenMP not investigated (not functional or beneficial)

- Curie: not clear how to explicitly distribute processes.
- Huygens: run with 240 IFS, 64 NEMO, 16 OASIS:
  - first 10 nodes with 2 OASIS and 30 IFS
  - last 2 nodes with 32 NEMO processes each



Maybe more benefit on clusters with limited bandwidth and for high resolution runs with 10x more processes.



Computing performance as a function of the number of computing cores for different depopulated-core conditions and tasks placement conditions.

- Results courtesy of Albanne Lecointre
- depopulated nodes give best performance
- fully-populated nodes are the most efficient.
- round-robin better than collecting neighbouring domains on a node.

I/O of EC-EARTH

- run of low-resolution EC-EARTH, good filesystem, so no I/O bottleneck.
- However, metadata operations take majority of I/O time.
- all OASIS tasks read and write continuously.
- all NEMO tasks write continuously.

Improve efficiency of postprocessing of GRIB data

- at first it seemed that CDO was a bottleneck, but that is incorrect. CDO is quick, if compiled correctly.
- Tests on huygens confirmed this.

TODO

- Refine independent scaling analysis
- Distributing tasks on Curie nodes: Find out how and test with hi-res EC-EARTH (and maybe NEMO).
- Make I/O report of hi-res EC-EARTH on Curie.
- Investigate performance of conversion with CMOR2?

END

Undercommitting nodes on Curie

- it seems that normal and radiation steps are faster, but the I/O or coupling steps get slower.
- Undercommitting nodes is not beneficial on Curie.