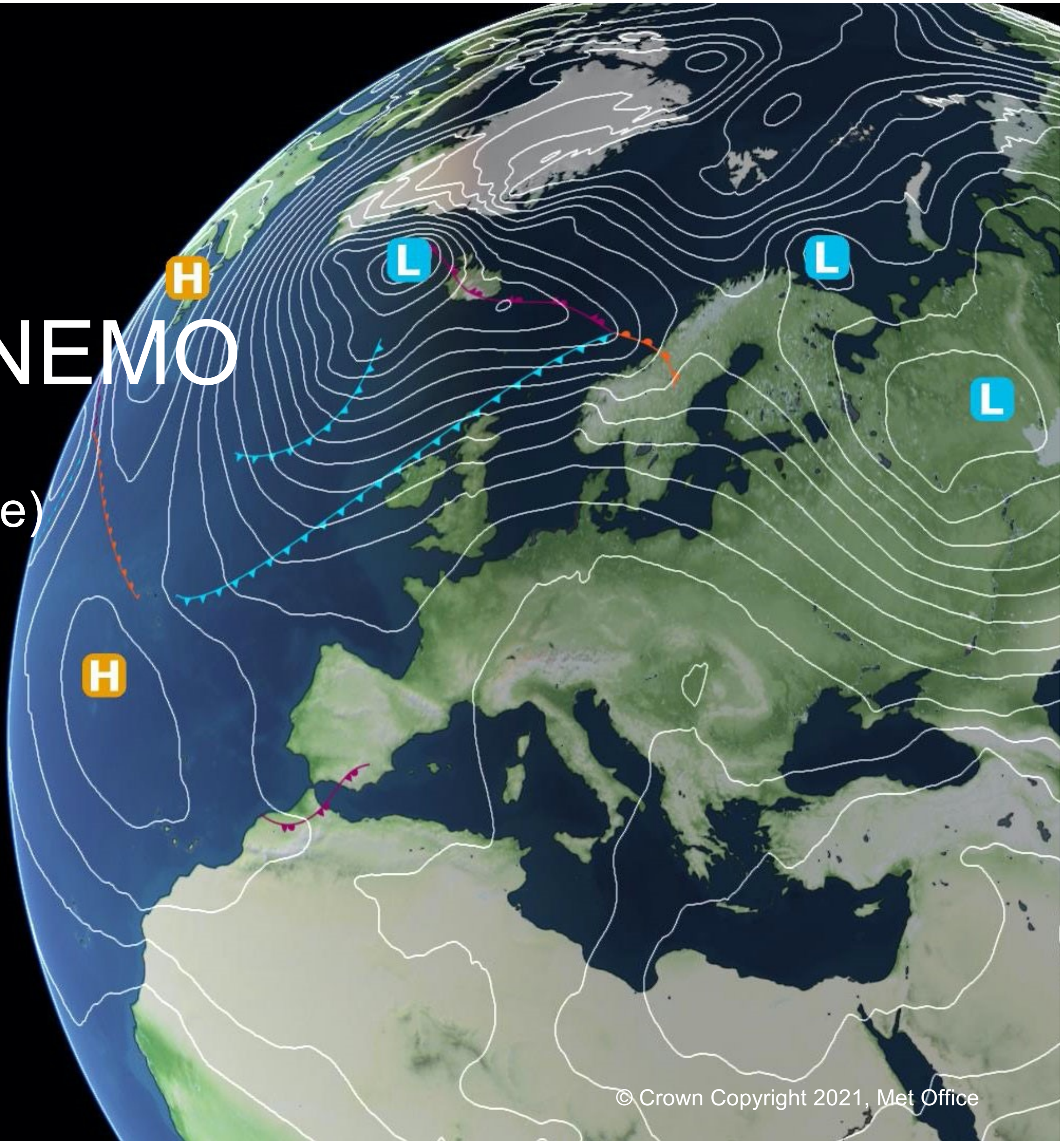
Met Office

Tiling performance in NEMO

D. Calvert, M. Bell, M. Glover (Met Office)

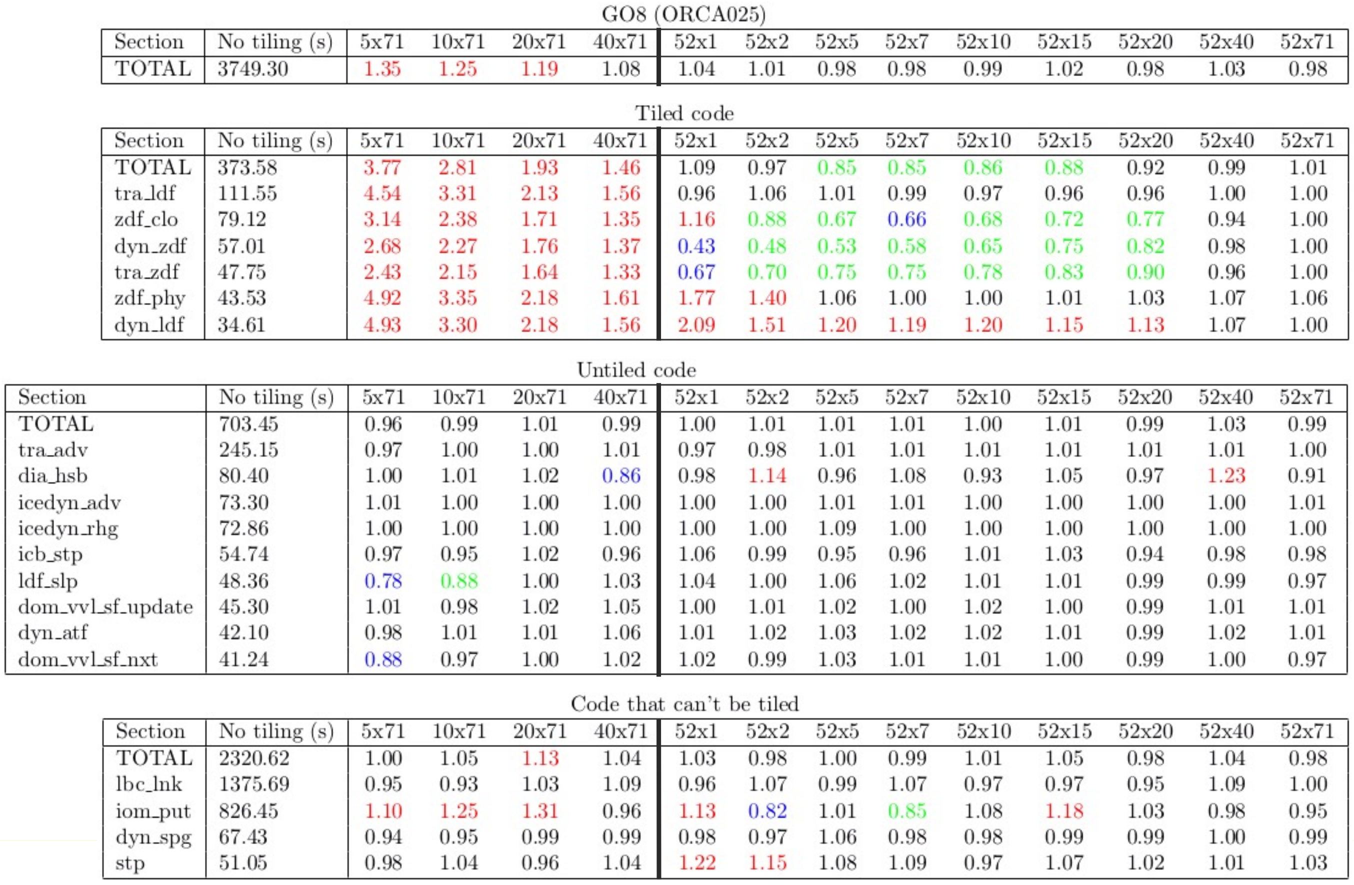
S. Masson, G. Madec (IPSL)

I. Epicoco, F. Mele (CMCC)



Met Office

Performance in ORCA025



GO8 (ORCA025)

Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	3749.30	1.35	1.25	1.19	1.08	1.04	1.01	0.98	0.98	0.99	1.02	0.98	1.03	0.98

Tiled code

Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	373.58	3.77	2.81	1.93	1.46	1.09	0.97	0.85	0.85	0.86	0.88	0.92	0.99	1.01
tra_ldf	111.55	4.54	3.31	2.13	1.56	0.96	1.06	1.01	0.99	0.97	0.96	0.96	1.00	1.00
zdf_clo	79.12	3.14	2.38	1.71	1.35	1.16	0.88	0.67	0.66	0.68	0.72	0.77	0.94	1.00
dyn_zdf	57.01	2.68	2.27	1.76	1.37	0.43	0.48	0.53	0.58	0.65	0.75	0.82	0.98	1.00
tra_zdf	47.75	2.43	2.15	1.64	1.33	0.67	0.70	0.75	0.75	0.78	0.83	0.90	0.96	1.00
zdf_phy	43.53	4.92	3.35	2.18	1.61	1.77	1.40	1.06	1.00	1.00	1.01	1.03	1.07	1.06
dyn_ldf	34.61	4.93	3.30	2.18	1.56	2.09	1.51	1.20	1.19	1.20	1.15	1.13	1.07	1.00

Untiled code

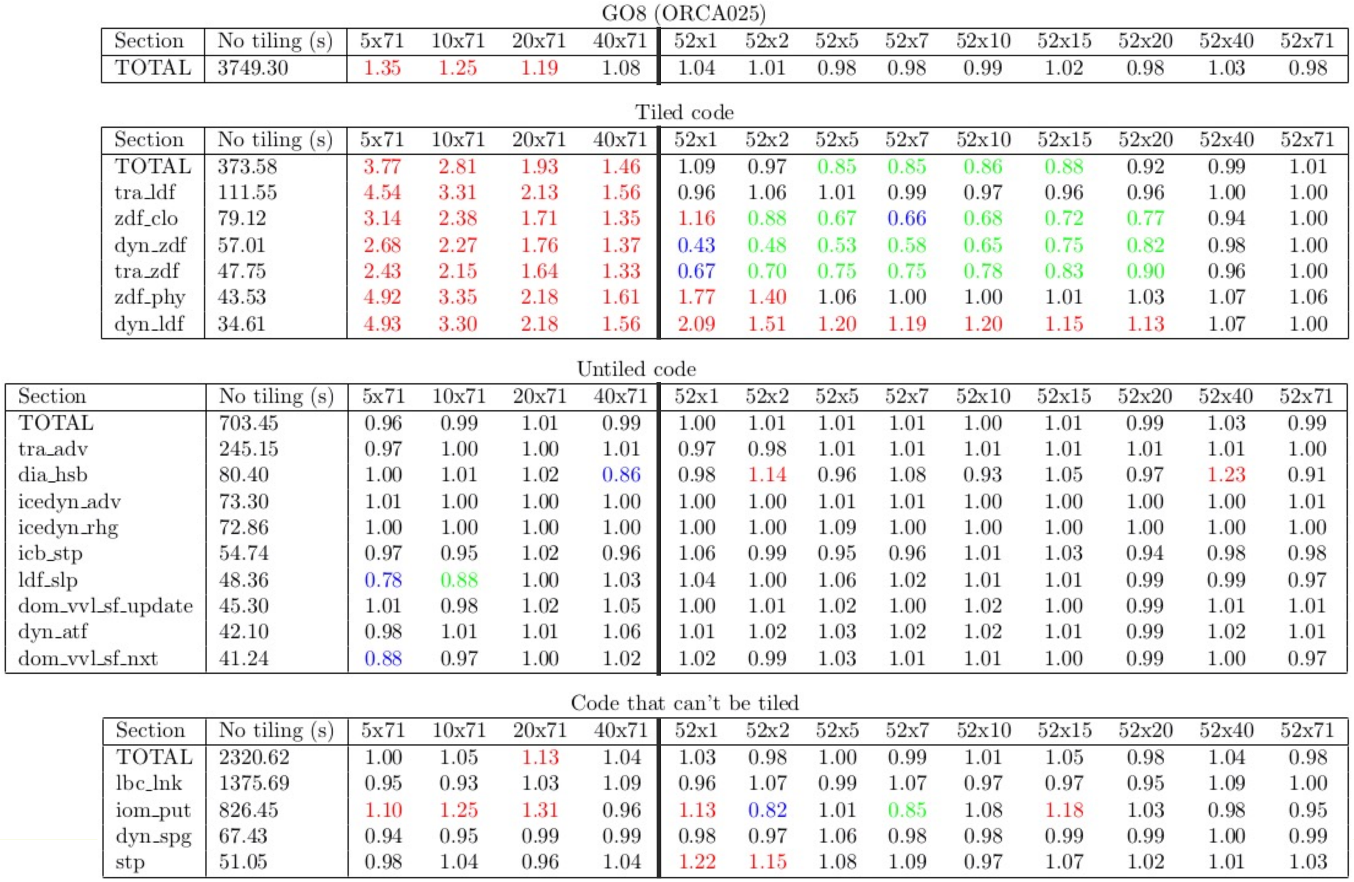
Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	703.45	0.96	0.99	1.01	0.99	1.00	1.01	1.01	1.01	1.00	1.01	0.99	1.03	0.99
tra_adv	245.15	0.97	1.00	1.00	1.01	0.97	0.98	1.01	1.01	1.01	1.01	1.01	1.01	1.00
dia_hsb	80.40	1.00	1.01	1.02	0.86	0.98	1.14	0.96	1.08	0.93	1.05	0.97	1.23	0.91
icedyn_adv	73.30	1.01	1.00	1.00	1.00	1.00	1.00	1.01	1.01	1.00	1.00	1.00	1.00	1.01
icedyn_rhg	72.86	1.00	1.00	1.00	1.00	1.00	1.00	1.09	1.00	1.00	1.00	1.00	1.00	1.00
icb_stp	54.74	0.97	0.95	1.02	0.96	1.06	0.99	0.95	0.96	1.01	1.03	0.94	0.98	0.98
ldf_slp	48.36	0.78	0.88	1.00	1.03	1.04	1.00	1.06	1.02	1.01	1.01	0.99	0.99	0.97
dom_vvl_sf_update	45.30	1.01	0.98	1.02	1.05	1.00	1.01	1.02	1.00	1.02	1.00	0.99	1.01	1.01
dyn_atf	42.10	0.98	1.01	1.01	1.06	1.01	1.02	1.03	1.02	1.02	1.01	0.99	1.02	1.01
dom_vvl_sf_nxt	41.24	0.88	0.97	1.00	1.02	1.02	0.99	1.03	1.01	1.01	1.00	0.99	1.00	0.97

Code that can't be tiled

Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	2320.62	1.00	1.05	1.13	1.04	1.03	0.98	1.00	0.99	1.01	1.05	0.98	1.04	0.98
lbc_lnk	1375.69	0.95	0.93	1.03	1.09	0.96	1.07	0.99	1.07	0.97	0.97	0.95	1.09	1.00
iom_put	826.45	1.10	1.25	1.31	0.96	1.13	0.82	1.01	0.85	1.08	1.18	1.03	0.98	0.95
dyn_spg	67.43	0.94	0.95	0.99	0.99	0.98	0.97	1.06	0.98	0.98	0.99	0.99	1.00	0.99
stp	51.05	0.98	1.04	0.96	1.04	1.22	1.15	1.08	1.09	0.97	1.07	1.02	1.01	1.03

- GO8 (eORCA025), NEMO4.2
  - 30d runs
  - `Ni_0, Nj_0 = (52, 71)`
  - `nn_hls = 2`, no QCO, no loop fusion
- `timing.output` sections vs tile size ("`ixj`"), as a fraction of time without tiling
  - `lbc_lnk` and `iom_put` added
  - `zdf_phy` split into closure scheme (`zdf_clo`) and other called subroutines
- Top ~90% of time spent in:
  - Tiled code (10%)
  - Untiled code (19%)
  - Code that can't be tiled (62%)
- Tiling in `i` is always slower
- Tiling in `j` has very little impact on overall times

Performance in ORCA025



GO8 (ORCA025)

Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	3749.30	1.35	1.25	1.19	1.08	1.04	1.01	0.98	0.98	0.99	1.02	0.98	1.03	0.98

Tiled code

Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	373.58	3.77	2.81	1.93	1.46	1.09	0.97	0.85	0.85	0.86	0.88	0.92	0.99	1.01
tra_ldf	111.55	4.54	3.31	2.13	1.56	0.96	1.06	1.01	0.99	0.97	0.96	0.96	1.00	1.00
zdf_clo	79.12	3.14	2.38	1.71	1.35	1.16	0.88	0.67	0.66	0.68	0.72	0.77	0.94	1.00
dyn_zdf	57.01	2.68	2.27	1.76	1.37	0.43	0.48	0.53	0.58	0.65	0.75	0.82	0.98	1.00
tra_zdf	47.75	2.43	2.15	1.64	1.33	0.67	0.70	0.75	0.75	0.78	0.83	0.90	0.96	1.00
zdf_phy	43.53	4.92	3.35	2.18	1.61	1.77	1.40	1.06	1.00	1.00	1.01	1.03	1.07	1.06
dyn_ldf	34.61	4.93	3.30	2.18	1.56	2.09	1.51	1.20	1.19	1.20	1.15	1.13	1.07	1.00

Untiled code

Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	703.45	0.96	0.99	1.01	0.99	1.00	1.01	1.01	1.01	1.00	1.01	0.99	1.03	0.99
tra_adv	245.15	0.97	1.00	1.00	1.01	0.97	0.98	1.01	1.01	1.01	1.01	1.01	1.01	1.00
dia_hsb	80.40	1.00	1.01	1.02	0.86	0.98	1.14	0.96	1.08	0.93	1.05	0.97	1.23	0.91
icedyn_adv	73.30	1.01	1.00	1.00	1.00	1.00	1.00	1.01	1.01	1.00	1.00	1.00	1.00	1.01
icedyn_rhg	72.86	1.00	1.00	1.00	1.00	1.00	1.00	1.09	1.00	1.00	1.00	1.00	1.00	1.00
icb_stp	54.74	0.97	0.95	1.02	0.96	1.06	0.99	0.95	0.96	1.01	1.03	0.94	0.98	0.98
ldf_slp	48.36	0.78	0.88	1.00	1.03	1.04	1.00	1.06	1.02	1.01	1.01	0.99	0.99	0.97
dom_vvl_sf_update	45.30	1.01	0.98	1.02	1.05	1.00	1.01	1.02	1.00	1.02	1.00	0.99	1.01	1.01
dyn_atf	42.10	0.98	1.01	1.01	1.06	1.01	1.02	1.03	1.02	1.02	1.01	0.99	1.02	1.01
dom_vvl_sf_nxt	41.24	0.88	0.97	1.00	1.02	1.02	0.99	1.03	1.01	1.01	1.00	0.99	1.00	0.97

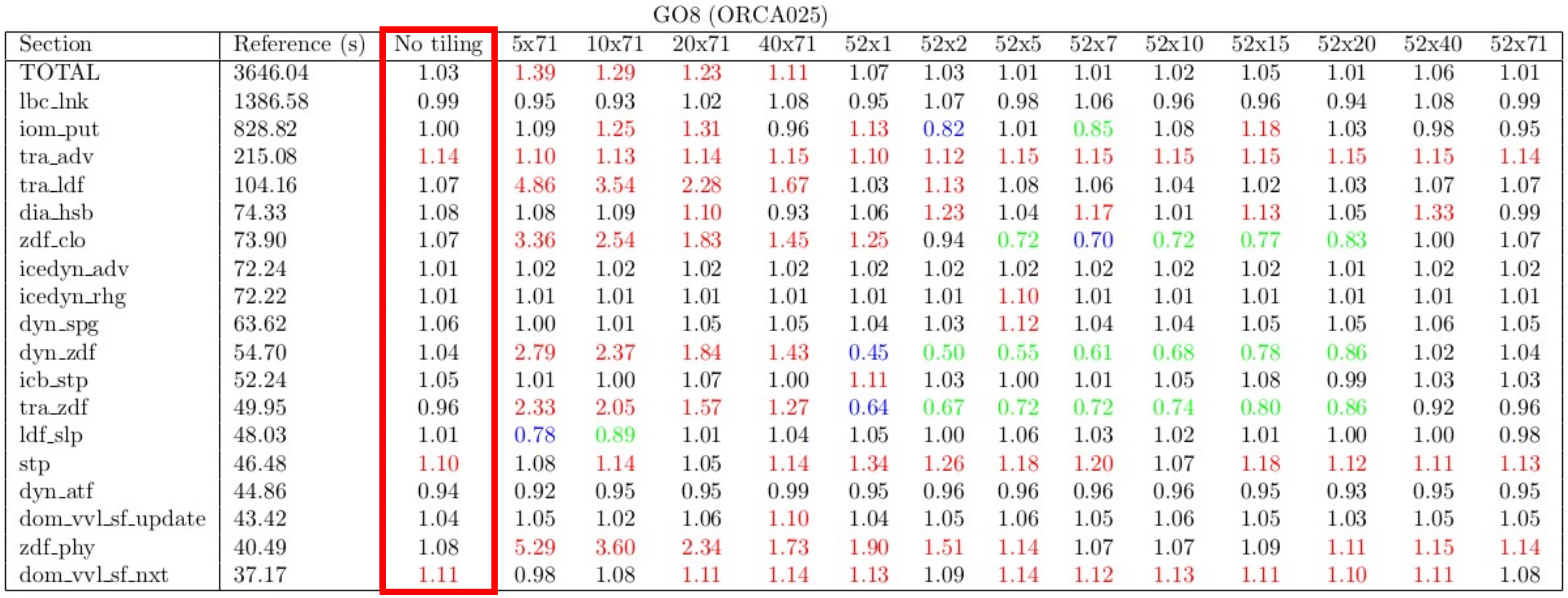
Code that can't be tiled

Section	No tiling (s)	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	2320.62	1.00	1.05	1.13	1.04	1.03	0.98	1.00	0.99	1.01	1.05	0.98	1.04	0.98
lbc_lnk	1375.69	0.95	0.93	1.03	1.09	0.96	1.07	0.99	1.07	0.97	0.97	0.95	1.09	1.00
iom_put	826.45	1.10	1.25	1.31	0.96	1.13	0.82	1.01	0.85	1.08	1.18	1.03	0.98	0.95
dyn_spg	67.43	0.94	0.95	0.99	0.99	0.98	0.97	1.06	0.98	0.98	0.99	0.99	1.00	0.99
stp	51.05	0.98	1.04	0.96	1.04	1.22	1.15	1.08	1.09	0.97	1.07	1.02	1.01	1.03

- The impact of tiling varies with code
  - `tra_ldf` and `dyn_ldf` scale poorly with tile size compared to ORCA2
  - `zdf_clo`, `zdf_phy` and `dyn_ldf` slow down at small tile sizes, but `dyn_zdf` and `tra_zdf` speed up
  - Some other inexpensive code (e.g. `tra_qsr`) is slowed down by the tiling
- Tiling isn't fully implemented
  - The FCT scheme (`tra_adv`) requires `nn_hls=3`
  - VVL will be replaced by QCO (tiled & cheaper)
  - Much of the remaining code is unlikely to benefit from the tiling (SI3 is mostly 2D)
  - However, using QCO and turning off much of this code (ICB, `dia_hsb`, SI3) has little impact
- There are performance issues unrelated to the tiling
  - Severe load imbalance (`lbc_lnk`)
  - Halo data is sent to XIOS (`iom_put`)
  - These times are reduced in more recent versions of NEMO (by ~90s and ~400s)

Performance in ORCA025

- Times as a fraction of the time with `nn_hls = 1` and no tiling (“Reference”)
- Tiling has to work against the cost of `nn_hls = 2`



GO8 (ORCA025)

Section	Reference (s)	No tiling	5x71	10x71	20x71	40x71	52x1	52x2	52x5	52x7	52x10	52x15	52x20	52x40	52x71
TOTAL	3646.04	1.03	1.39	1.29	1.23	1.11	1.07	1.03	1.01	1.01	1.02	1.05	1.01	1.06	1.01
lbc_lnk	1386.58	0.99	0.95	0.93	1.02	1.08	0.95	1.07	0.98	1.06	0.96	0.96	0.94	1.08	0.99
iom_put	828.82	1.00	1.09	1.25	1.31	0.96	1.13	0.82	1.01	0.85	1.08	1.18	1.03	0.98	0.95
tra_adv	215.08	1.14	1.10	1.13	1.14	1.15	1.10	1.12	1.15	1.15	1.15	1.15	1.15	1.15	1.14
tra_ldf	104.16	1.07	4.86	3.54	2.28	1.67	1.03	1.13	1.08	1.06	1.04	1.02	1.03	1.07	1.07
dia_hsb	74.33	1.08	1.08	1.09	1.10	0.93	1.06	1.23	1.04	1.17	1.01	1.13	1.05	1.33	0.99
zdf_clo	73.90	1.07	3.36	2.54	1.83	1.45	1.25	0.94	0.72	0.70	0.72	0.77	0.83	1.00	1.07
icedyn_adv	72.24	1.01	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.01	1.02	1.02
icedyn_rhg	72.22	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.10	1.01	1.01	1.01	1.01	1.01	1.01
dyn_spg	63.62	1.06	1.00	1.01	1.05	1.05	1.04	1.03	1.12	1.04	1.04	1.05	1.05	1.06	1.05
dyn_zdf	54.70	1.04	2.79	2.37	1.84	1.43	0.45	0.50	0.55	0.61	0.68	0.78	0.86	1.02	1.04
icb_stp	52.24	1.05	1.01	1.00	1.07	1.00	1.11	1.03	1.00	1.01	1.05	1.08	0.99	1.03	1.03
tra_zdf	49.95	0.96	2.33	2.05	1.57	1.27	0.64	0.67	0.72	0.72	0.74	0.80	0.86	0.92	0.96
ldf_slp	48.03	1.01	0.78	0.89	1.01	1.04	1.05	1.00	1.06	1.03	1.02	1.01	1.00	1.00	0.98
stp	46.48	1.10	1.08	1.14	1.05	1.14	1.34	1.26	1.18	1.20	1.07	1.18	1.12	1.11	1.13
dyn_atf	44.86	0.94	0.92	0.95	0.95	0.99	0.95	0.96	0.96	0.96	0.96	0.95	0.93	0.95	0.95
dom_vvl_sf_update	43.42	1.04	1.05	1.02	1.06	1.10	1.04	1.05	1.06	1.05	1.06	1.05	1.03	1.05	1.05
zdf_phy	40.49	1.08	5.29	3.60	2.34	1.73	1.90	1.51	1.14	1.07	1.07	1.09	1.11	1.15	1.14
dom_vvl_sf_nxt	37.17	1.11	0.98	1.08	1.11	1.14	1.13	1.09	1.14	1.12	1.13	1.11	1.10	1.11	1.08

Met Office

Halo calculations & tiling

- For a 2D loop over an MPI domain with internal size (*X*, *Y*), halo width *H* and tile size (*x*, *y*), the total number of loop iterations *N* scales as:

$$\frac{N}{XY} = 1 + 2\,\mathrm{H}\,\frac{x+y}{xy} + \frac{4H^2}{xy}$$

- Local working arrays: not preserved in memory, so must calculate all points on a tile
  - Calculations depend on tile and halo size
- Module / allocatable arrays: preserved in memory, so no need to repeat calculations done by other tiles (`DO_?D_OVR` macros)
  - Calculations depend only on halo size

```
DO_2D( 1, 1, 1, 1 )
   zwrk(ji,jj) = 0.
END_2D
```

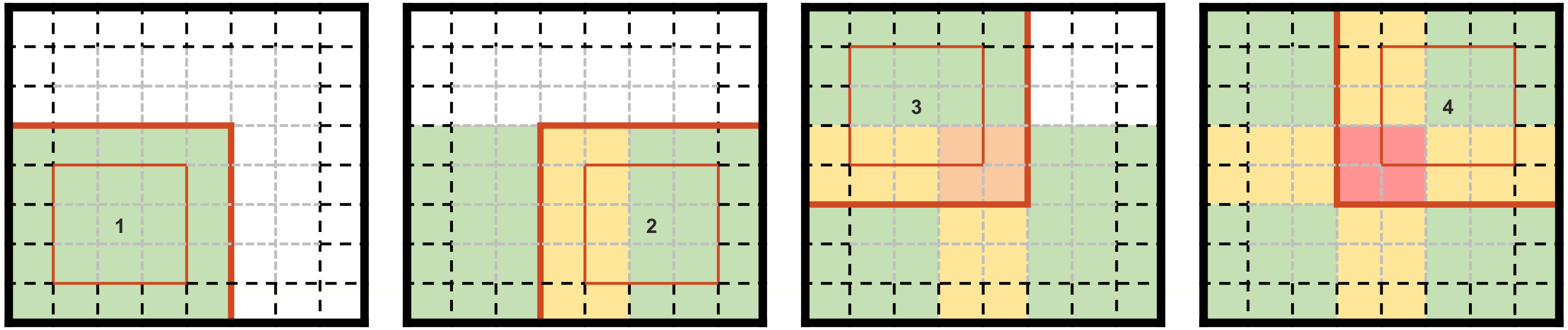
⬜ Undefined

🟩 1 calculation

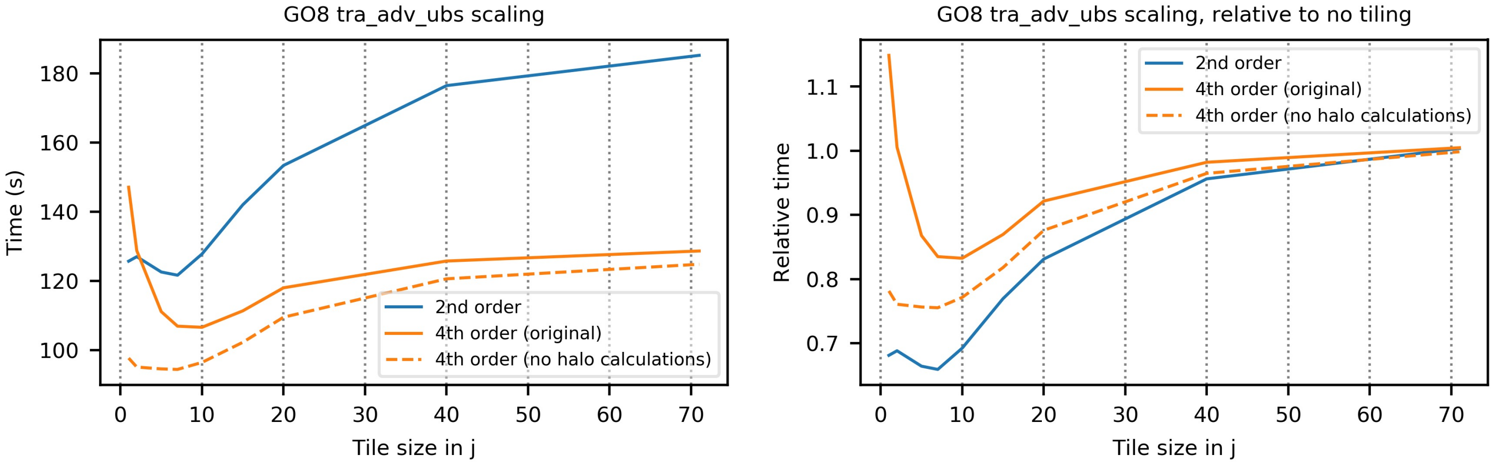
🟨 2 calculations

🟧 3 calculations

🟥 4 calculations

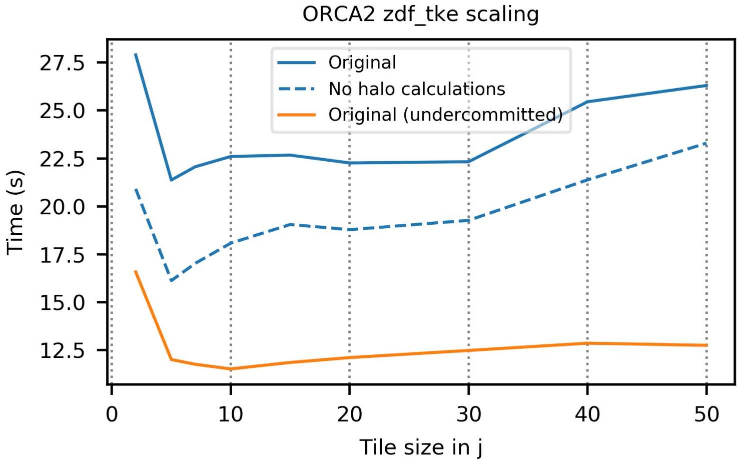


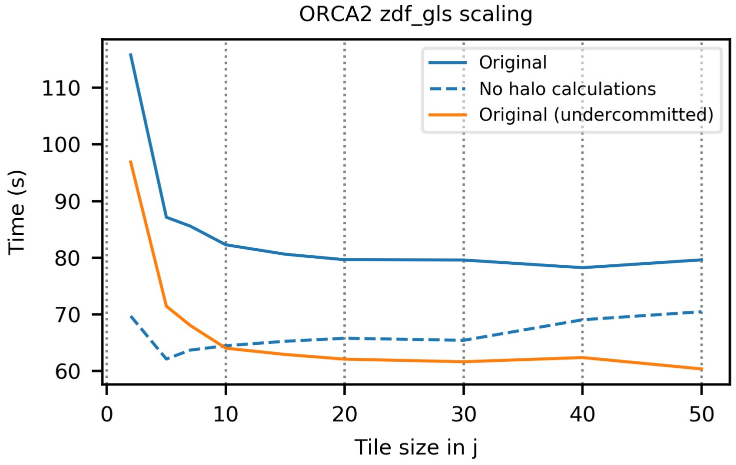
Remove unnecessary halo calculations



- The 4th order UBS scheme uses `interp_4th_cpt` from `tra_adv_fct`
- Removing the unnecessary halo calculations from this subroutine:
  - Improves time without tiling
  - Improves scaling with tile size

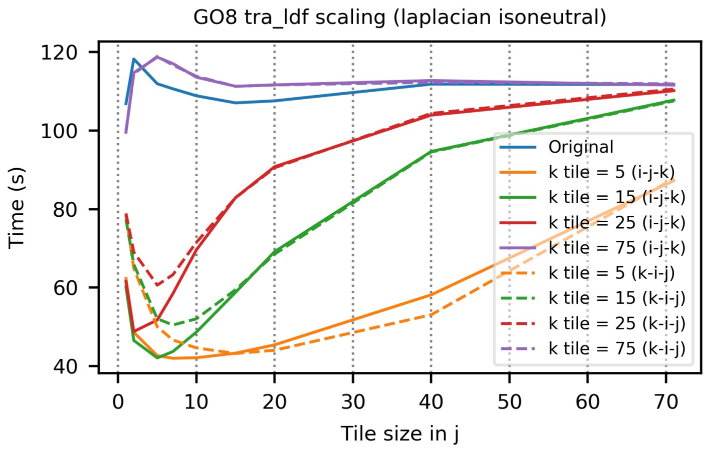
Replace halo calculations with `lbc_lnk`

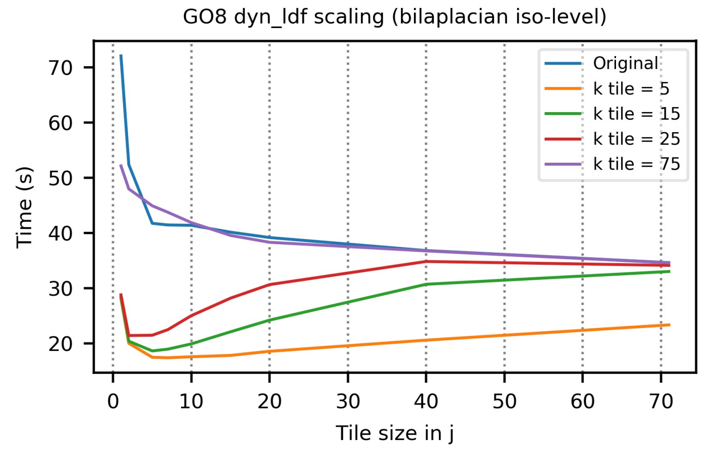


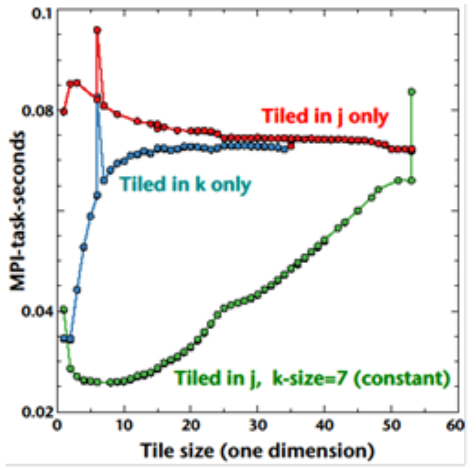


- Using less CPUs per node socket increases the effective memory bandwidth per CPU
  - The time penalty of cache misses is reduced; rough estimate of potential tiling impact
  - Tiling in `zdf_tke` and `zdf_gls` should be able to perform better
- ZDF closures are purely 1D, but `avm` is needed on haloes to calculate shear (`zdf_sh2`) and for `dyn_zdf`
- Reverting to using `lbc_lnk` instead of halo calculations:
  - Slightly improves `zdf_tke` scaling (30% faster vs 20%)
  - Improves `zdf_gls` scaling (tiling no longer slows the code down)
- `zdf_tke`/`zdf_gls` times without tiling are slower/faster when including the cost of the `lbc_lnk`

Tiling in the k dimension







Maff Glover, Met Office

- `tra_ldf` & `dyn_ldf` scale poorly in GO8 compared to ORCA2
- Larger domain size- more cache misses
  - `56i x 75j x 75k` (GO8)
  - `34i x 54j x 31k` (ORCA2)
- Maff Glover's work with a similar configuration shows 3D tiling is needed to realise the full potential of tiling
- Tiling in k recovers the ORCA2 scaling (and improves on it)
- Fitting data in cache is more important than ensuring contiguous memory access

Met Office

Summary

- Overall tiling performance is poor in an eORCA025 reference configuration (GO8)
  - Time is mostly spent in code that can’t be tiled- separate issue (load imbalance, `iom_put` haloes)
  - Tiling is not implemented everywhere- not much scope for improvement here (except FCT scheme)
  - Some tiled code is underperforming- optimisation is possible
- Halo calculations reduce the performance of timestep-level tiling
  - Remove unnecessary halo calculations
  - Replace halo calculations with `lbc_lnk` calls (but which is worse for performance/scalability?)
- 2D tiling can limit performance
  - Tiling in `k` is necessary to ensure consistent performance between configurations