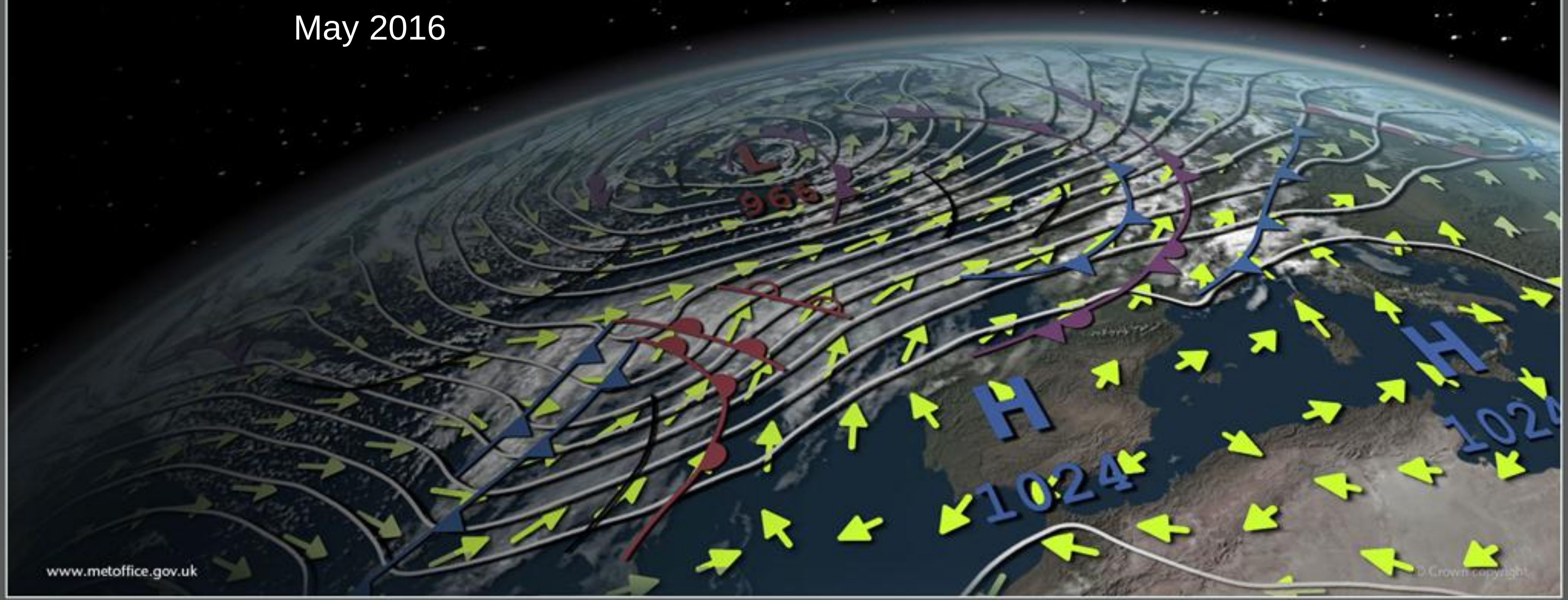
Met Office

Stitching Threads into the Unified Model

M. J. Glover, A. J. Malcolm, M. Guidolin and P. Selwood

May 2016



Overview

- The Unified Model
- Targets for OpenMP
- Threading performance
- Resource stealing
- Summary

The Unified Model

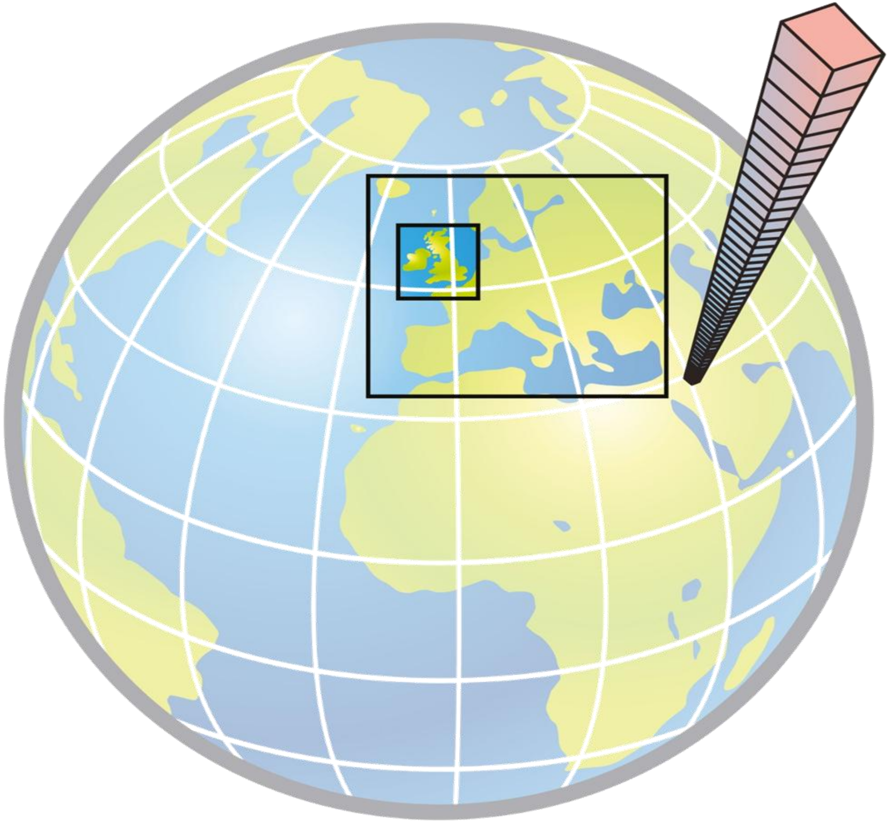
Unified Model technical

UM Atmosphere

- Over 20 years old
- Fortran / MPI / OpenMP
- Global Collaboration
- Rapidly changing

Coupled Systems

- 4DVAR Assimilation
- UKCA Atmospheric Chemistry
- NEMO Ocean



Amdahl and threading

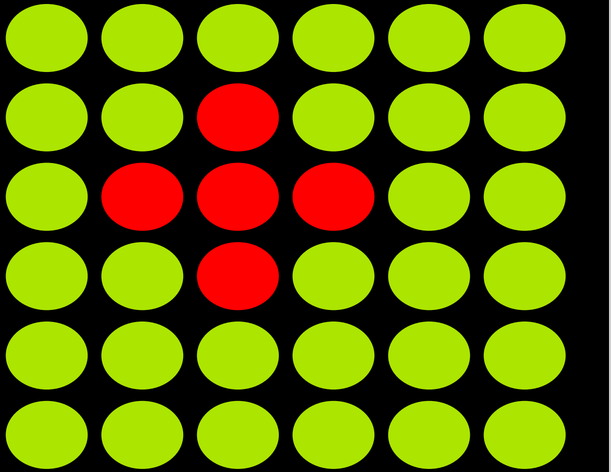
$$t = t_s + \frac{t_p}{n}$$

- UM has no particular hotspots
- Requires large OpenMP code coverage

Data dependencies

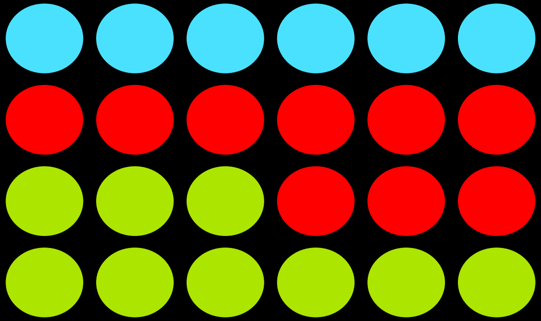
- Finite difference
- Hard-coded dependencies between neighbours

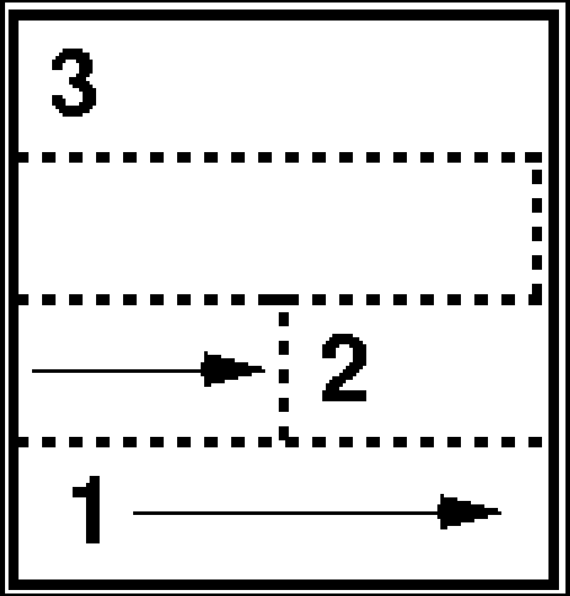
=> Many loop-level directives.



Segmentation

- Used for radiation, convection + microphysics.
- No horizontal dependencies
- Tune size for cache
- OpenMP parallel





Targets for OpenMP

DrHook

```
IF(lhook) call dr_hook(
‘MODULE:ROUTINE’,
      0,   zhook_handle)
```

- Written at ECMWF (S. Saarinen, M. Hamrud, D. Salmond, J. Hague)
- Callipers stay in the code
- Threadsafe
- Parameter `lhook`: no production impact

Other tools?

CrayPat? Score-P / Scalasca?

DrHook:

- Same tool and output across platforms
- Text-based output easily post-processed
- Potential for MPMD: coupled models

Scaling Score

$$S = \frac{\left(\dfrac{p_1 t_1 T_1}{p_2 t_2 T_2} - 1 \right)}{\left(\dfrac{p_1 t_1}{p_2 t_2} - 1 \right)} + 1$$

p : MPI tasks
t : threads

T : elapsed time

1. Perform baseline run
2. Increase number of threads
3. Run comparison script

Scaling Score

Score value	Meaning
S < 0	Antiscaling
S = 0	No scaling
0 < S < 1	Sub-linear scaling
S = 1	Ideal scaling
S > 1	Super-linear scaling

Use in addition to wallclock profile.

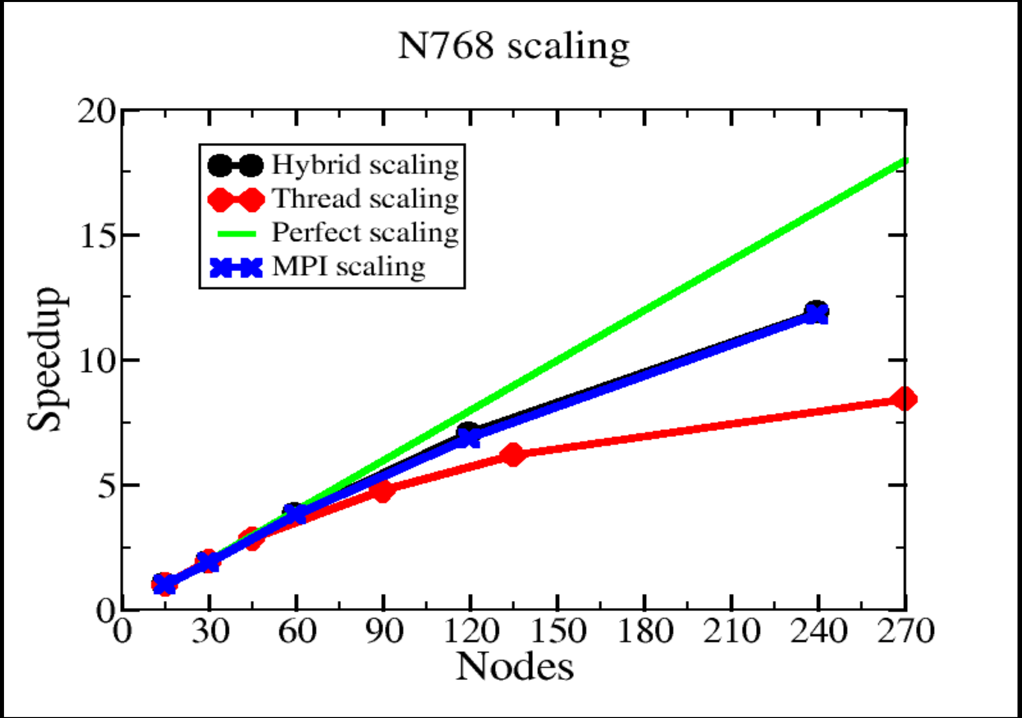
Threading Performance

Comparison of MPI and thread space (Operational global configuration)

Curves have common baseline resource.

Hybrid: best performing combination.

Demonstrates coverage of threading.

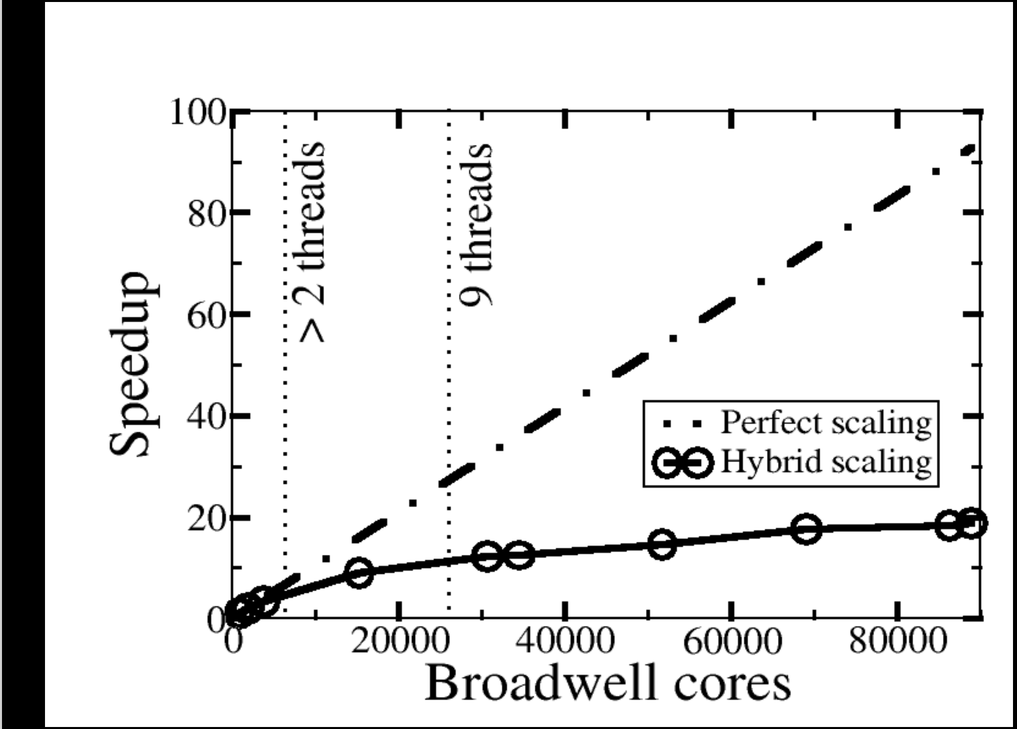


Operational configuration

3 threads has become more efficient than 2 for given resource.

Nodes	Cores	Threads	Time (s)	Perfect scaling	Hybrid scaling
15	540	2	1137	1	1
30	1080	2	1137	2	1.89
60	2160	2	564	4	3.81
120	4320	3	305	8	7.05
240	8640	3	181	16	11.88
480	17280	3	111	32	19.37

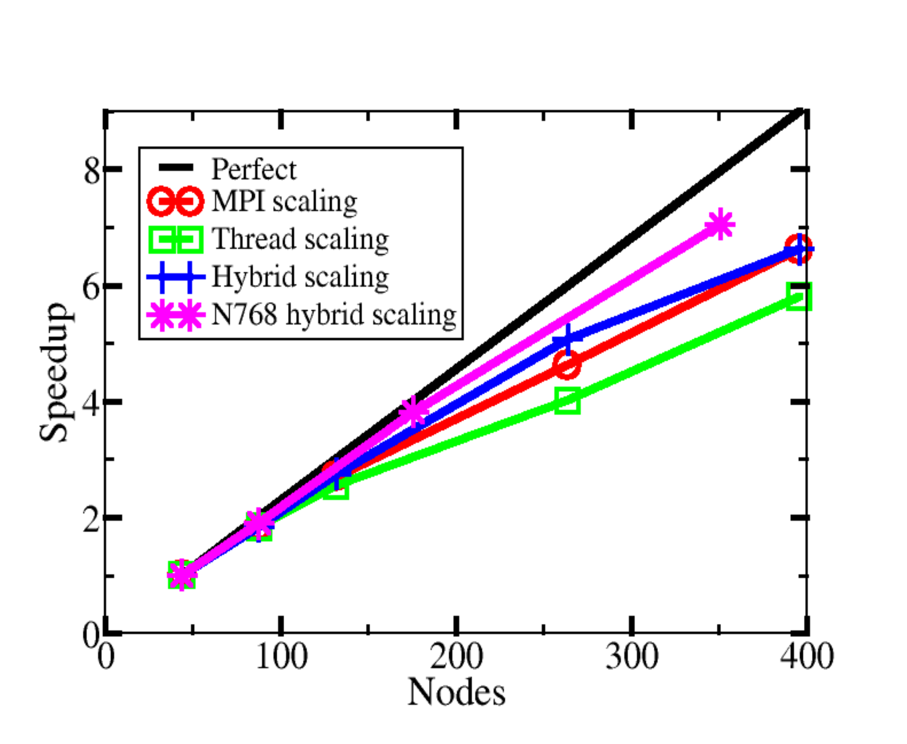
Hybrid parallelisation at scale



More than 2 threads required to avoid MPI-turnover at ~10k cores.

Hybrid scales to over 80k cores.

Limited Area Model (LAM)



Global & LAM: adjusted to ~ same gridpoints per MPI task.

Hybrid out-scales MPI and thread scaling for LAM.

Scaling less good than global, despite no poles!

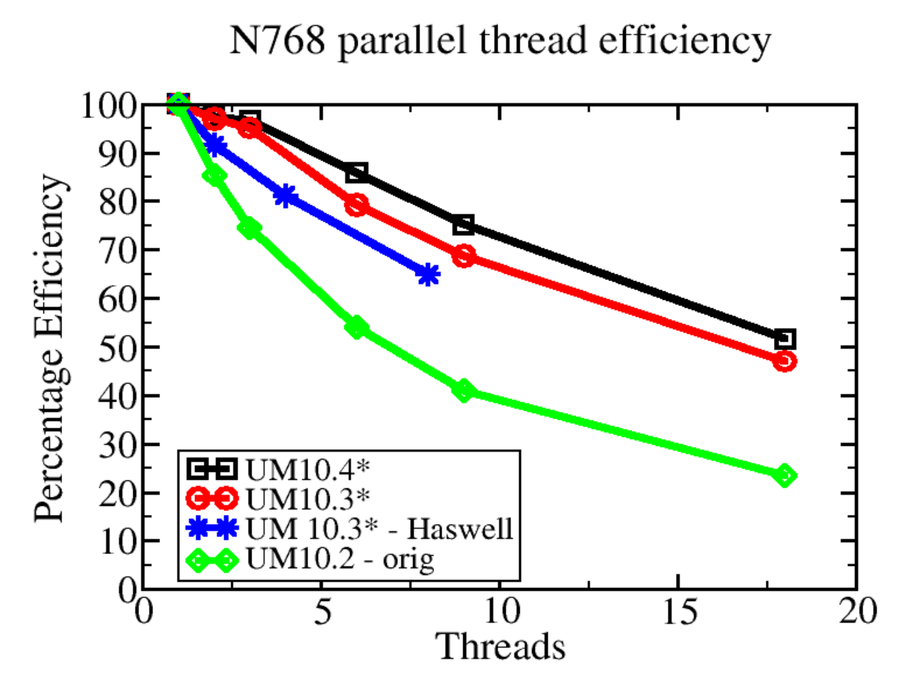
Met Office

Progress over model versions

8 threads:
~40% to ~75%.

UM 10.2: mid-2015
UM 10.4*: early 2016
(10.x* *just* predates 10.x)

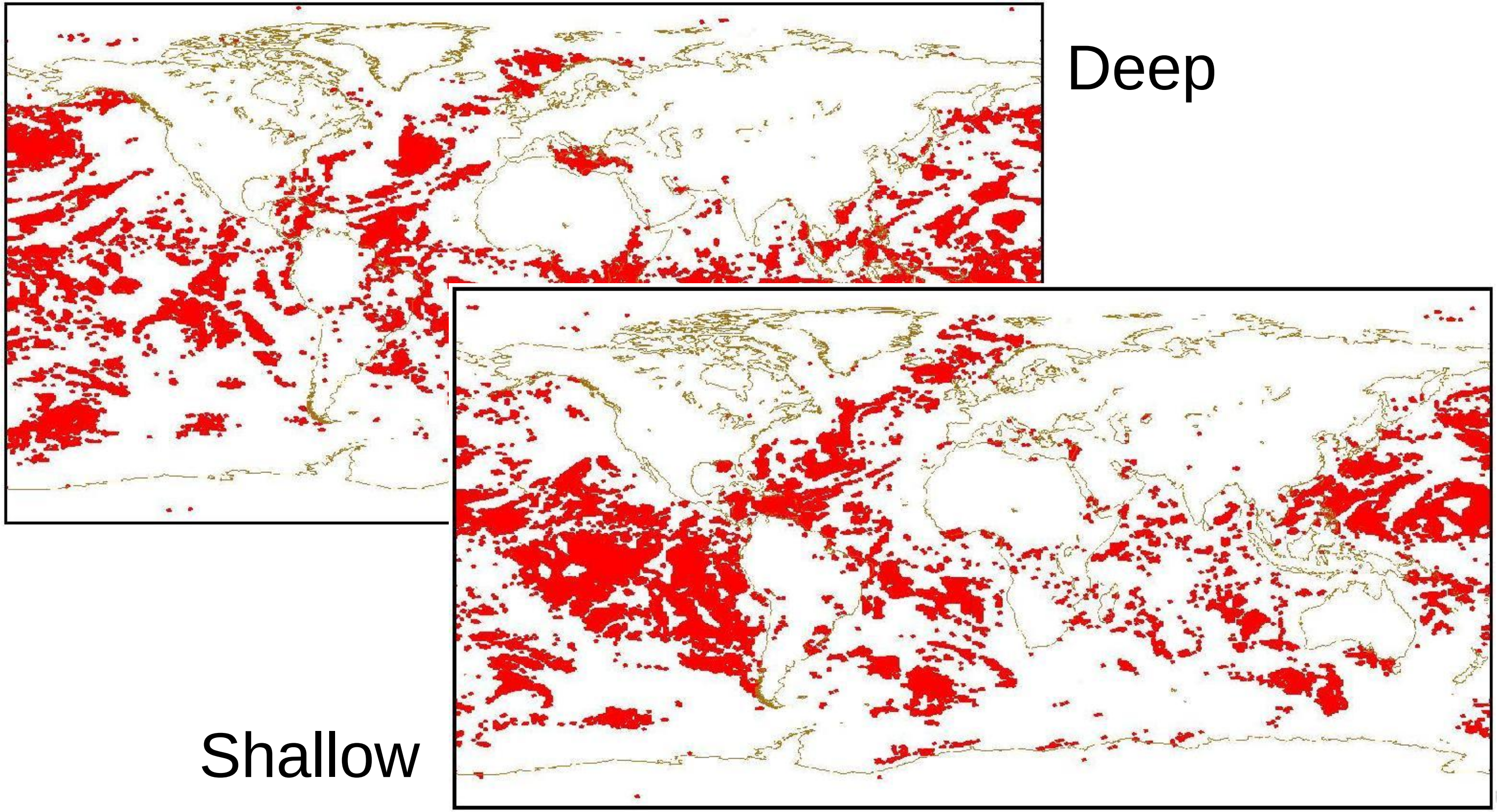
Boost from Haswell to Broadwell. Better shared L3 cache?



Resource-stealing

Malleable thread counts

Load imbalance: convection

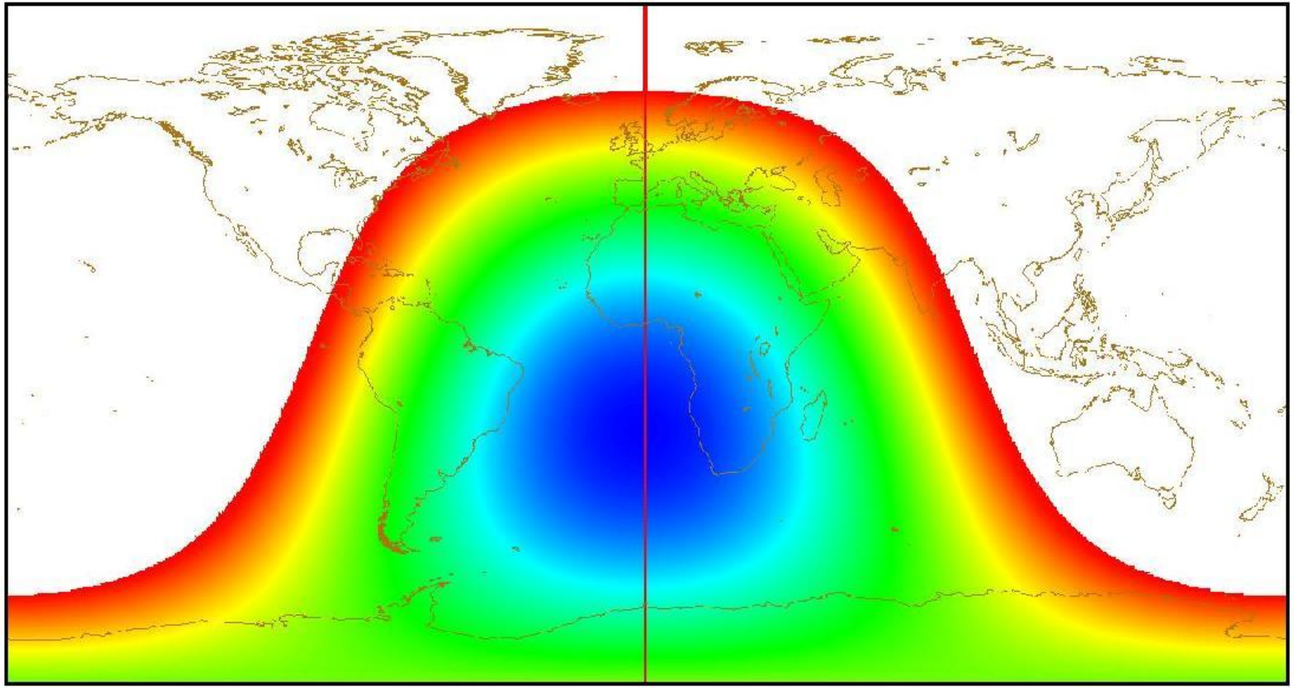


Deep

Shallow

Load imbalance: short wave radiation

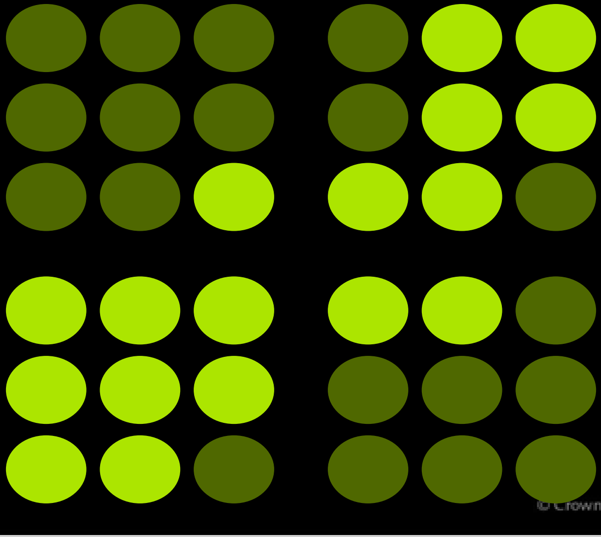
Incoming flux (January)



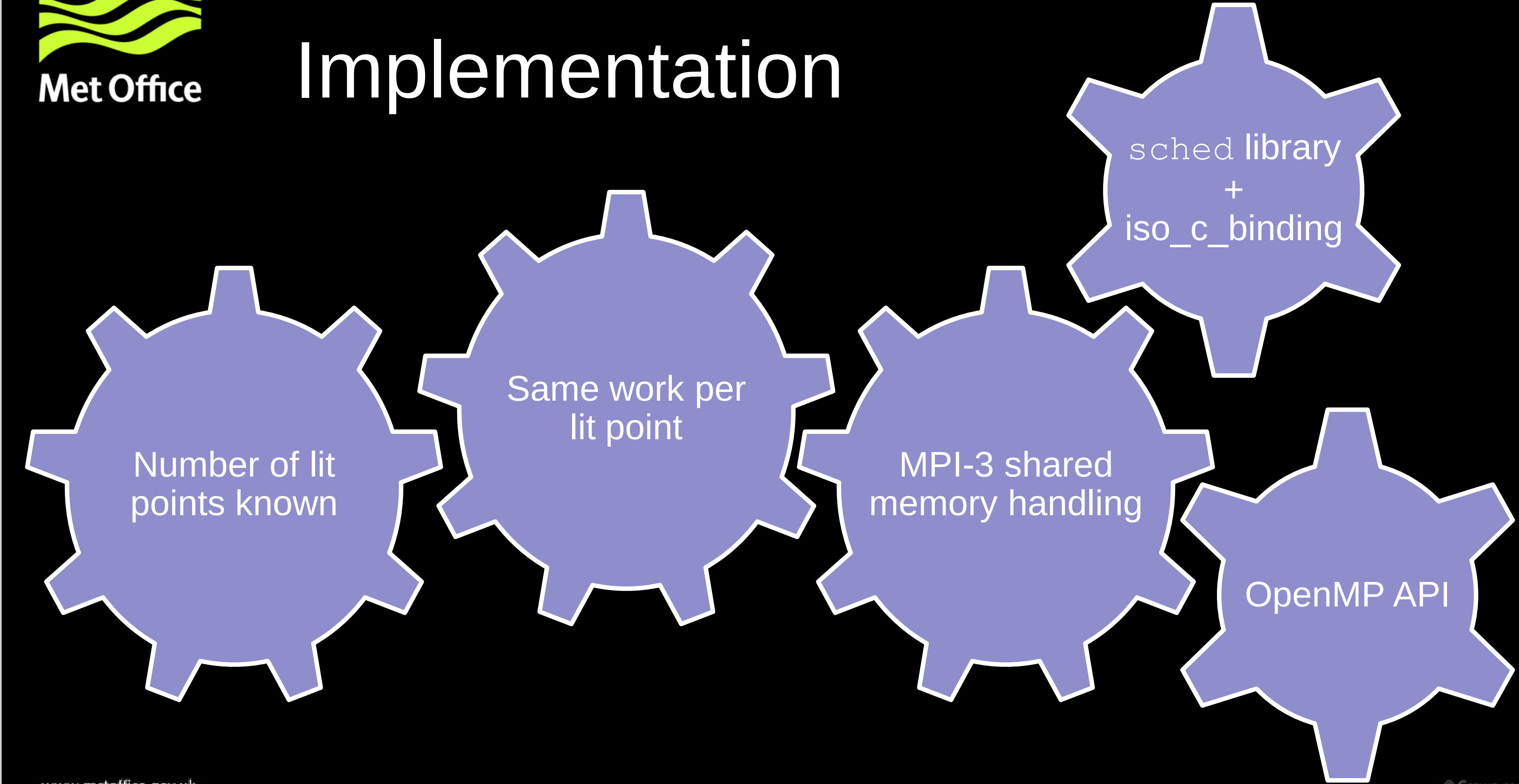
Resource-stealing

More work, more cores. (And more threads!)

Shortwave radiation



Implementation



Number of lit points known

Same work per lit point

MPI-3 shared memory handling

`sched` library + iso_c_binding

OpenMP API

Shared memory arrays

- Each thread stores its core ID
- Once at the start of a run

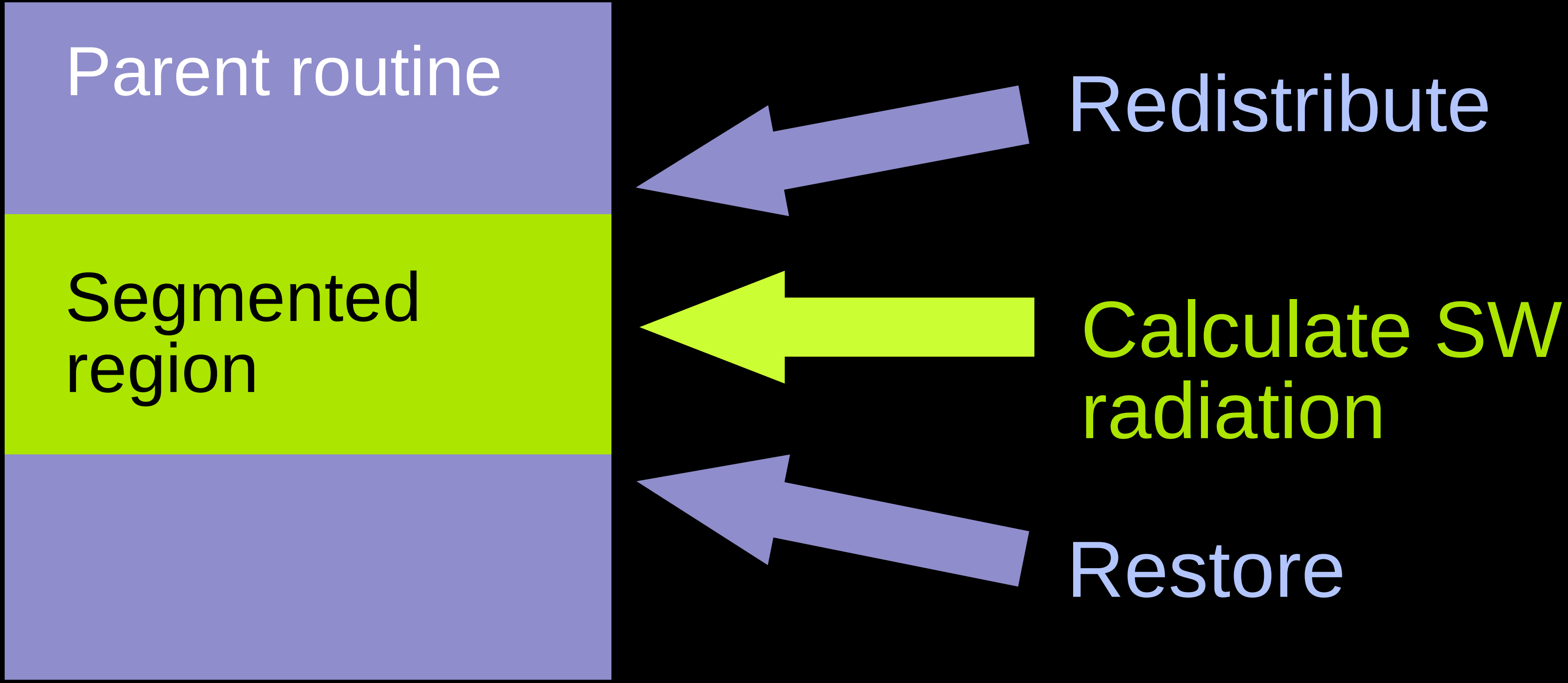
```
aprun -cc cpu
```

```
1:threads*MPI tasks
```

- MPI reduction on-node to find total work
- Each task calculates and stores number of threads it needs

```
1:MPI tasks
```

Redistribution



Parent routine

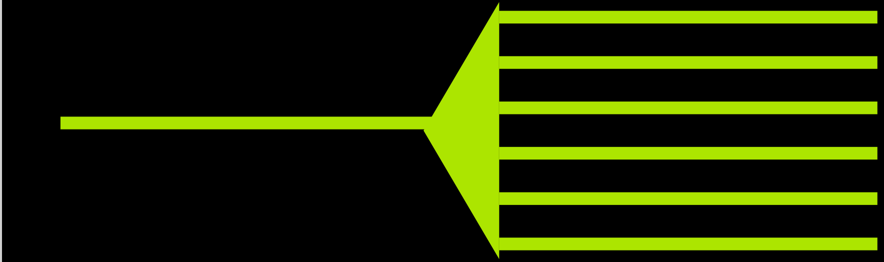
Segmented region

Redistribute

Calculate SW radiation

Restore

Obstacles: thread pooling

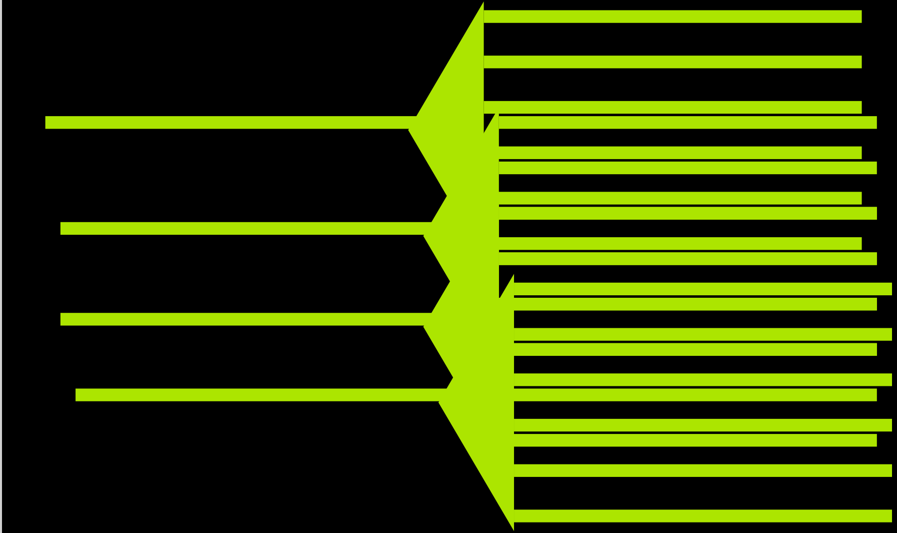


- Threads spawned but not destroyed.
- Threads kept for next parallel region.

Good when number of threads is constant or fewer (concurrency throttling).

Met Office

Obstacles: thread pooling



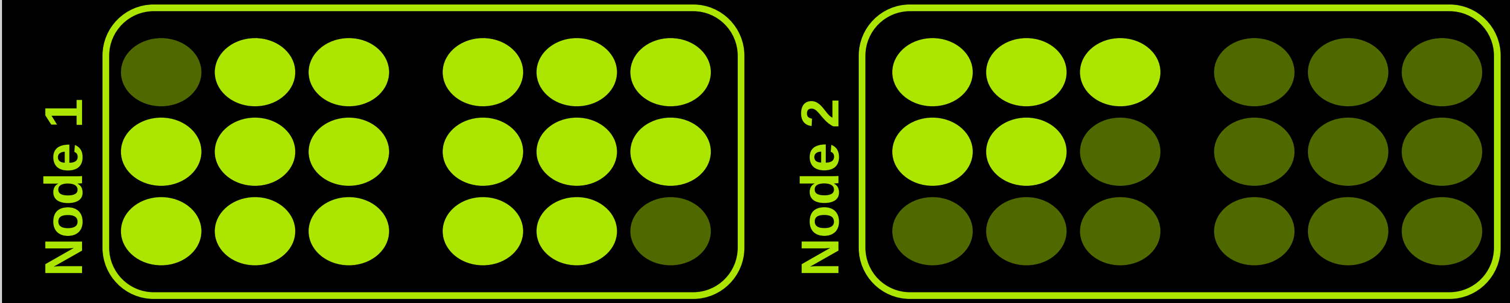
- Compiler version `>= cce/8.4.0`
- Earlier cce: affinitize to virtual cores.

- **Passive** OMP wait policy for entire model
- Changing with Cray API causes hang

Obstacles: spatial correlation

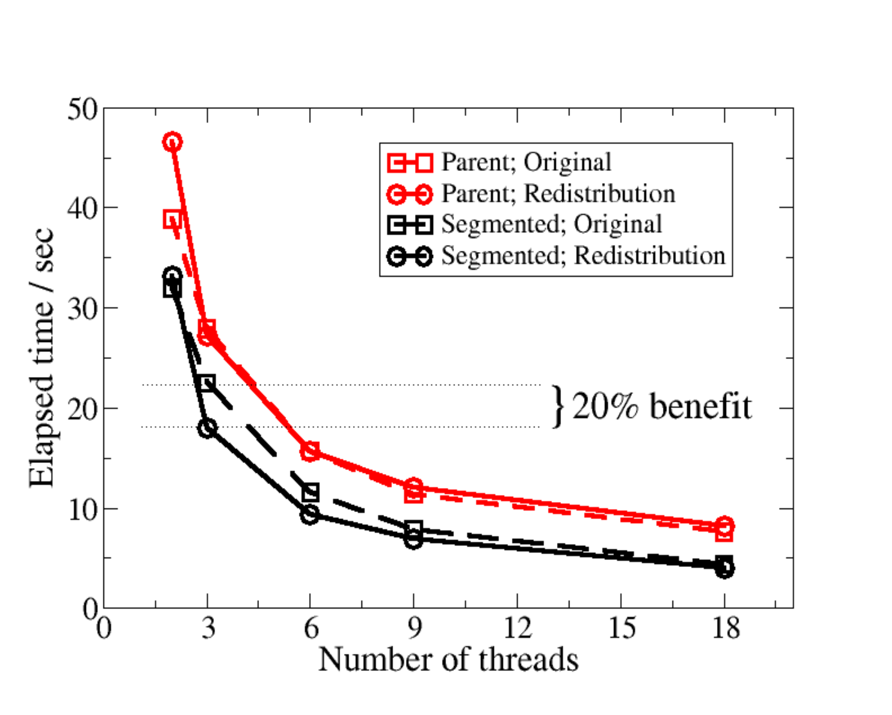
Scatter high-workload MPI-tasks between nodes: Round-Robin ordering.

```
MPICH_RANK_REORDER_METHOD=0
```



Default SMP-style

Redistribution timings



Segmented region: performance improves on all thread counts >= 3.

Parent routine: performance degrades. Cache invalidation?

Summary

Summary

UM hybrid performance

- Largely loop-level due to data dependencies
- Needs high coverage
- Benefits realised in operations and at scale

Resource stealing

- Up to 20% benefit on calculation itself, but ...
- Large overheads.
- Thread pooling: may have adverse impacts.

Met Office

Questions?