UNIVERSITY OF LEEDS

# Computational Efficiency of the Aerosol Scheme in the Met Office Unified Model

*Mark Richardson, Ph.D.*

*CEMAC, University of Leeds*

*The MONSooN system, a collaborative facility supplied under the Joint Weather and Climate Research Programme, a strategic partnership between the Met Office and the Natural Environment Research Council, was used for this work.*

# Background

UNIVERSITY OF LEEDS

- UM has functionality for aerosol and chemistry by way of UKCA
  - UKCA uses GLOMAP for aerosol processes and ASAD for chemistry
  - Users of UKCA mainly model climate although some want to use with AQM
- Resolution
  - UK Met. Office Unified Model (UM)
  - All this work is with N96 (192x144x85 grid boxes) ~ 2degree
- MPI configuration
  - 128 MPI tasks in 2D topology (16x8)
- Various OpenMP
- My job was funded by JWCRP (NCAS) in collaboration with Met. Office
  - To assess computational efficiency and address places in code where it appears inefficient
  - Introduce OpenMP to UKCA for enhanced parallelism
  - The UKCA version of GLOMAP has no OpenMP so already planned to introduce it for enhanced parallelism (some history as done in 2010 with academic code TOMCAT)
- First work done 2015 – 2016 with vn8.6
  - Slightly ahead of academia and slightly behind climate group at UKMO
  - Migrate to vn10.x to lodge on trunk
  - This was managed a vn10.6 branch (DONE Sep2016) except for OpenMP

# Motivation

UNIVERSITY OF LEEDS

- When activated UKCA adds overhead
  - Configuration 1, full chemistry and aerosol
    - 30% of run is >40% overhead
  - Configuration 2: reduced chemistry
    - with pre-calculated concentrations, i.e. offline oxidants
    - 20% of run is ~25% overhead
- Climate simulations – as stated by their scientists
  - Years 1850-1950 and 1950-2050 (also pre-industrial, non-anthropogenic scenarios)
  - At best 15 months per day
  - Resolution is N96L85 (192x144x85 grid boxes)
  - Expect to use 448 cpus to achieve
- Options
  - Reduced complexity e.g. reduced GLOMAP and fewer chemical species
  - more parameterisation (some groups are doing this)
  - Limit the frequency of calculation, typical only call UKCA every third time step
  - Improve computational efficiency (a continuous evaluation)

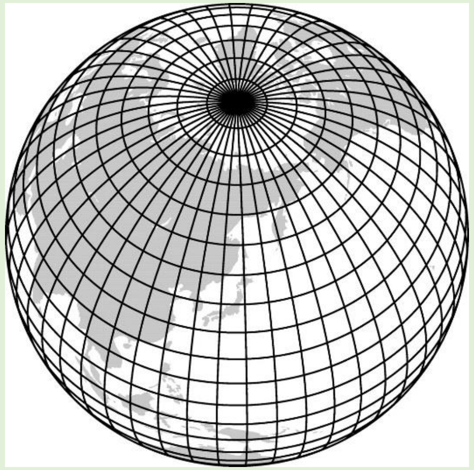
# Acknowledgements

UNIVERSITY OF LEEDS

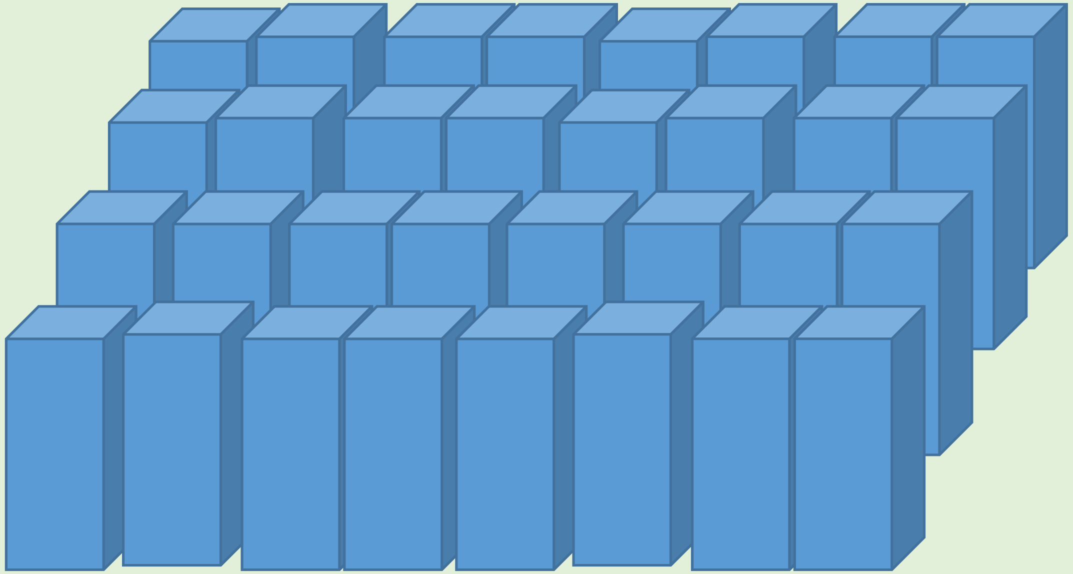

- NCAS funding in association with NERC, https://www.ncas.ac.uk/
- JWCRP collaboration with UK Met Office, http://www.jwcrp.org.uk/
- University of Leeds School of Earth and Environment for hosting the researcher http://see.leeds.ac.uk/
- Graham Mann, NCAS and School of Earth and Environment, University of Leeds
- Fiona O’Connor, Earth Systems and Mitigation Science, UK Met. Office
- Paul Selwood, HPC Optimisation, UK Met. Office
- UK Met Office Collaborative Service for access to MONSooN HPC system
  - A Cray XC40 reached through a secured gateway

# Relating physical space, computational domain and arrangement in memory

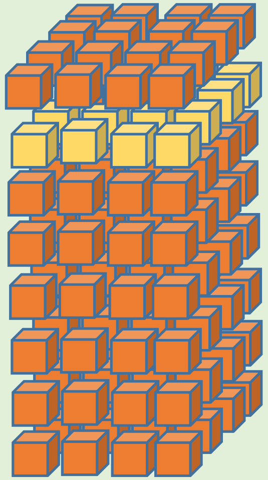
UNIVERSITY OF LEEDS



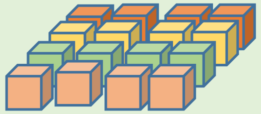
MPI 2D TOPOLOGY e.g. 8x4 MPI tasks



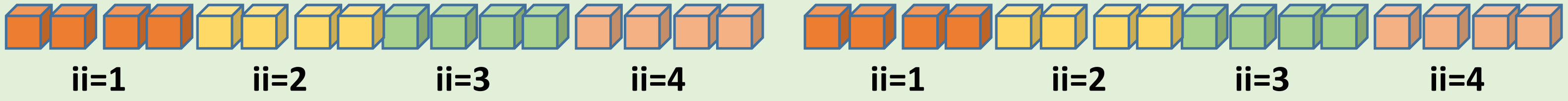
Consider one MPI task with
L = 1, model_levels



Examining one of those levels
ii=1,rows and kk=1, row_length



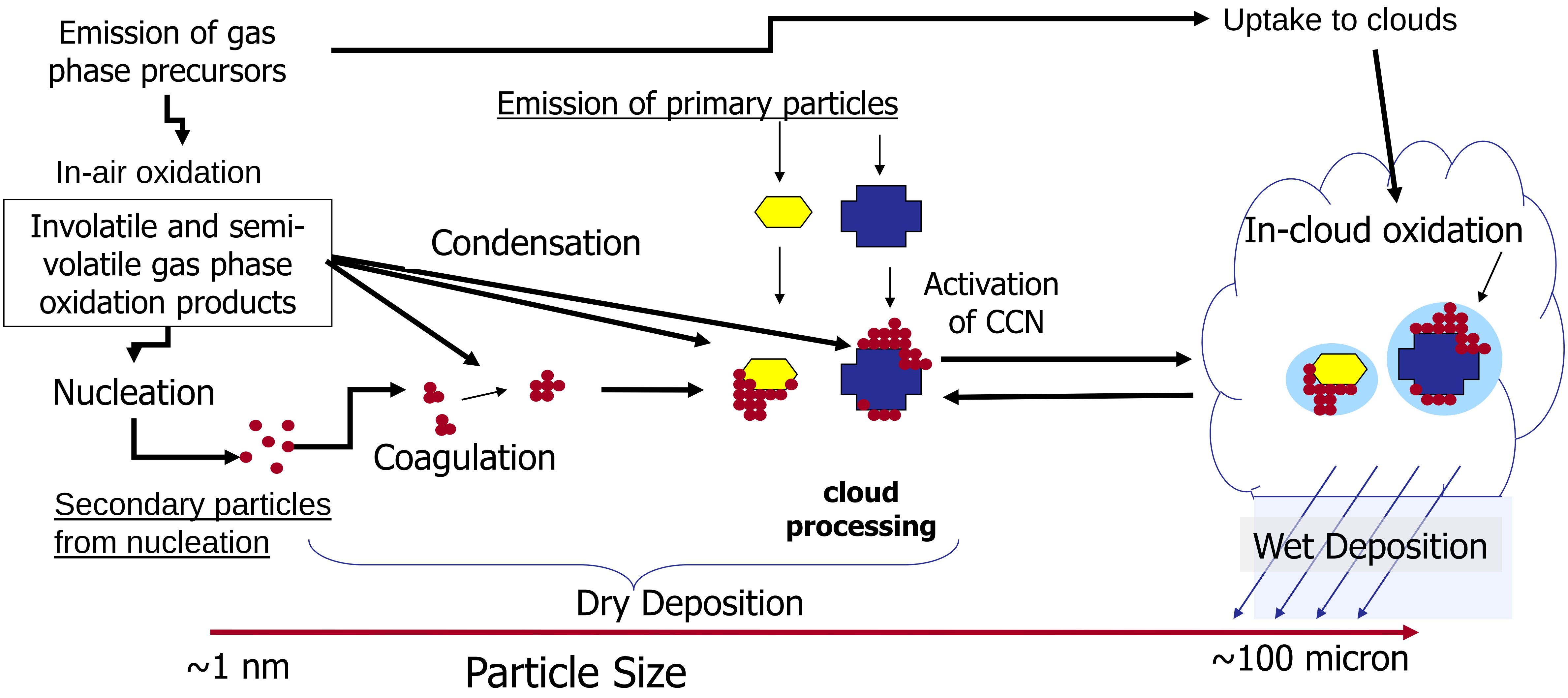
Two layers in memory






*Globe Graphic: Warren M Washington et al. Phil. Trans. R. Soc. A 2009;367:833-846*

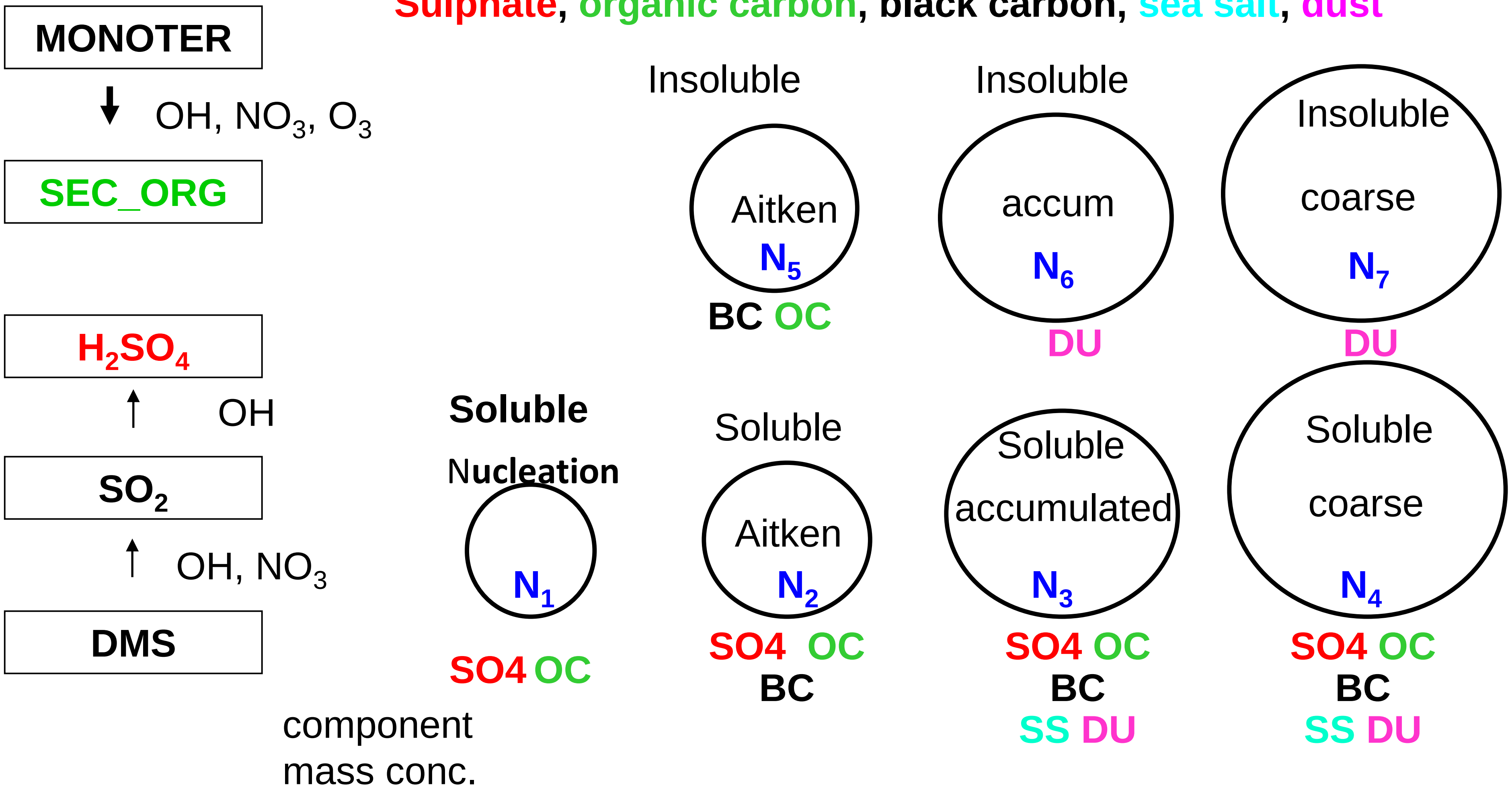
# Aerosol processes, size and composition



# GLOMAP-mode standard configuration (with dust)

UNIVERSITY OF LEEDS

**Sulphate, organic carbon, black carbon, sea salt, dust**




Aerosol mass as "components" in internally mixed modes

**19 mass**

**7 number**

**Transported tracers=26**

# DrHook reports

UNIVERSITY OF LEEDS

## Reference code 2 threads

Number of PEs: 128

**Header fields**

| | Min | Mean | Max | (Max-Min) |
|---|---|---|---|---|
| **Instrument overhead (%)** | 0.8 (PE 122) | 0.99 | 1.23 (PE 61) | 0.43 |
| **Heap (MB)** | 567 (PE 9) | 577.20 | 607 (PE 18) | 40 |
| **RSS (MB)** | 473 (PE 69) | 574.47 | 615 (PE 18) | 142 |
| **Stack (MB)** | 0 (PE 0) | 0.00 | 0 (PE 0) | 0 |
| **Paging** | 0 (PE 0) | 0.00 | 0 (PE 0) | 0 |
| **Wall Time (s)** | 399.33 (PE 124) | 401.47 | 407.91 (PE 95) | 8.58 |
| **Thread#1 (s)** | 399.3 (PE 9) | 399.31 | 399.31 (PE 0) | 0 |
| **Thread#2 (s)** | 22.14 (PE 115) | 36.42 | 52.64 (PE 40) | 30.50 |
| **Thread#1 (%)** | 97.89 (PE 95) | 99.46 | 99.99 (PE 107) | 2.10 |
| **Thread#2 (%)** | 5.54 (PE 115) | 9.07 | 13.03 (PE 40) | 7.49 |

**Ordering routines by self: mean**

| | Min | Mean | Max | (Max-Min) |
|---|---|---|---|---|
| **UKCA_*** | (PE ) | 140.70 | (PE ) | 0 |
| **TIMER@1** | 22.511 (PE 61) | 44.59 | 68.422 (PE 125) | 45.91 |
| **UKCA_COAGWITHNUCL@1** | 28.523 (PE 5) | 30.90 | 32.733 (PE 24) | 4.21 |
| **ATMOS_PHYSICS1@1** | 23.358 (PE 82) | 25.97 | 28.39 (PE 30) | 5.03 |
| **UKCA_ABDULRAZZAK_GHAN@1** | 10.16 (PE 108) | 22.68 | 29.909 (PE 70) | 19.75 |
| **UKCA_COND_COFF_V@1** | 15.662 (PE 118) | 17.50 | 19.009 (PE 23) | 3.35 |
| **HALO_EXCHANGE:SWAP_BOUNDS_NS_DP@1** | 10.071 (PE 125) | 15.36 | 20.534 (PE 72) | 10.46 |
| **UKCA_RADAER_BAND_AVERAGE@2** | 7.44 (PE 116) | 11.69 | 14.808 (PE 63) | 7.37 |
| **UKCA_RADAER_BAND_AVERAGE@1** | 7.485 (PE 124) | 11.57 | 14.414 (PE 61) | 6.93 |
| **eg_CUBIC_LAGRANGE@1** | 9.897 (PE 71) | 10.07 | 11.525 (PE 119) | 1.63 |
| **U_MODEL_4A@1** | 1.175 (PE 70) | 9.90 | 25.209 (PE 108) | 24.03 |
| **EG_CORRECT_TRACERS_UKCA@1** | 7.221 (PE 121) | 7.62 | 7.879 (PE 48) | 0.66 |
| **GLUE_CONV_6A@1** | 3.456 (PE 101) | 6.68 | 12.816 (PE 59) | 9.36 |
| **GLUE_CONV_6A@2** | 3.355 (PE 108) | 6.63 | 12.65 (PE 77) | 9.29 |
| **UM_WRITDUMP@1** | 0.482 (PE 0) | 6.45 | 6.751 (PE 6) | 6.27 |
| **EG_INTERPOLATION_ETA@1** | 4.232 (PE 71) | 6.30 | 9.533 (PE 122) | 5.30 |
| **HALO_EXCHANGE:SWAP_BOUNDS_EW_DP@1** | 5.801 (PE 121) | 6.22 | 6.568 (PE 90) | 0.77 |
| **SCATTER_FIELD_MPL@1** | 1.644 (PE 0) | 6.20 | 6.64 (PE 108) | 5 |
| **UKCA_CONDEN@1** | 4.916 (PE 115) | 6.02 | 6.555 (PE 61) | 1.64 |

## Development code 2 threads

Number of PEs: 128

**Header fields**

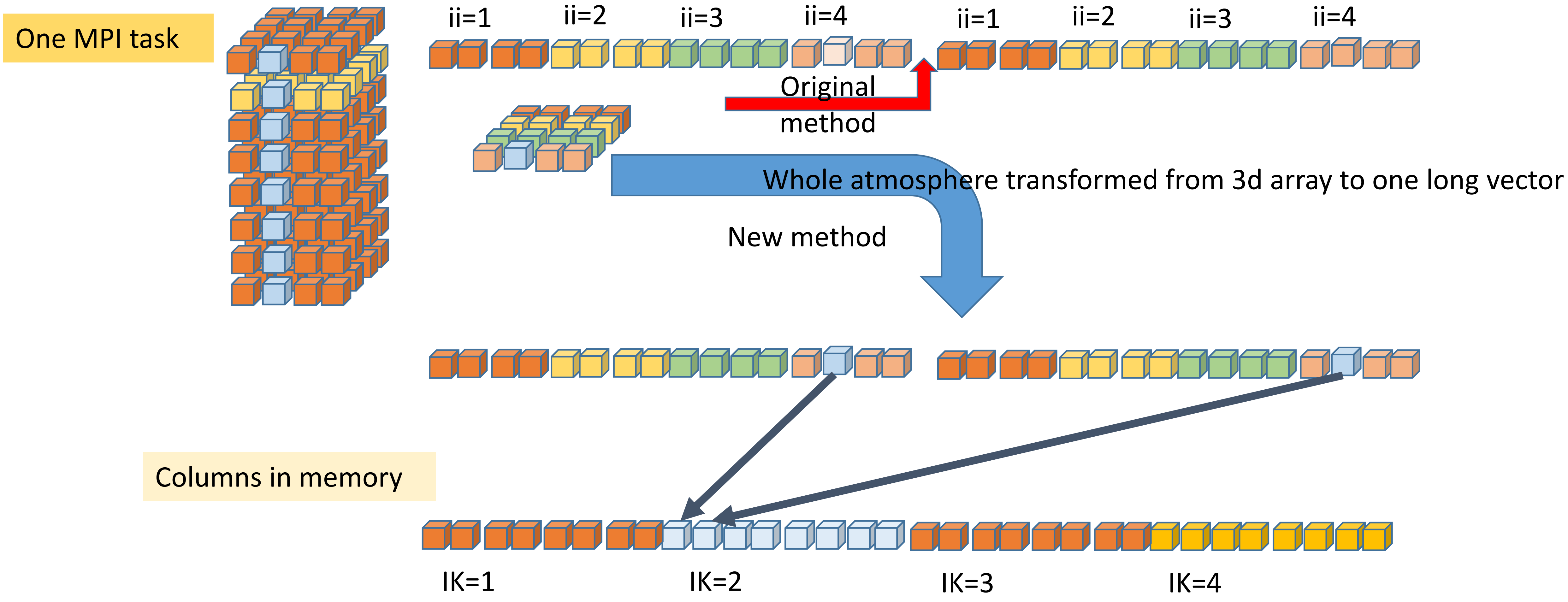
| | Min | Mean | Max | (Max-Min) |
|---|---|---|---|---|
| **Instrument overhead (%)** | 0.97 (PE 121) | 1.18 | 1.44 (PE 61) | 0.47 |
| **Heap (MB)** | 1391 (PE 112) | 1403.03 | 1430 (PE 40) | 39 |
| **RSS (MB)** | 501 (PE 112) | 525.65 | 563 (PE 40) | 62 |
| **Stack (MB)** | 0 (PE 0) | 0.00 | 0 (PE 0) | 0 |
| **Paging** | 0 (PE 0) | 0.00 | 0 (PE 0) | 0 |
| **Wall Time (s)** | 343.48 (PE 107) | 345.88 | 352.42 (PE 95) | 8.94 |
| **Thread#1 (s)** | 343.44 (PE 0) | 343.44 | 343.44 (PE 0) | 0 |
| **Thread#2 (s)** | 49.7 (PE 123) | 65.70 | 78.93 (PE 46) | 29.23 |
| **Thread#1 (%)** | 97.45 (PE 95) | 99.30 | 99.99 (PE 107) | 2.54 |
| **Thread#2 (%)** | 14.46 (PE 123) | 18.99 | 22.61 (PE 46) | 8.15 |

**Ordering routines by self: mean**

| | Min | Mean | Max | (Max-Min) |
|---|---|---|---|---|
| **UKCA_*** | (PE ) | 118.98 | (PE ) | 0 |
| **TIMER@1** | 24.25 (PE 49) | 44.68 | 66.561 (PE 125) | 42.31 |
| **ATMOS_PHYSICS1@1** | 23.491 (PE 59) | 25.96 | 28.447 (PE 49) | 4.96 |
| **UKCA_ABDULRAZZAK_GHAN@1** | 10.483 (PE 108) | 22.61 | 29.431 (PE 71) | 18.95 |
| **HALO_EXCHANGE:SWAP_BOUNDS_NS_DP@1** | 9.884 (PE 13) | 15.45 | 20.501 (PE 111) | 10.62 |
| **UKCA_RADAER_BAND_AVERAGE@2** | 7.485 (PE 116) | 11.76 | 14.873 (PE 63) | 7.39 |
| **UKCA_RADAER_BAND_AVERAGE@1** | 7.488 (PE 124) | 11.58 | 14.488 (PE 64) | 7 |
| **eg_CUBIC_LAGRANGE@1** | 9.884 (PE 19) | 10.06 | 11.45 (PE 119) | 1.57 |
| **UKCA_COND_COFF_V@1** | 7.415 (PE 119) | 8.41 | 9.099 (PE 43) | 1.68 |
| **UKCA_COND_COFF_V@2** | 7.584 (PE 97) | 8.41 | 9.147 (PE 58) | 1.56 |
| **UKCA_COAGWITHNUCL@2** | 7.872 (PE 116) | 8.13 | 8.468 (PE 85) | 0.60 |
| **UKCA_COAGWITHNUCL@1** | 7.82 (PE 2) | 8.10 | 8.403 (PE 81) | 0.58 |
| **U_MODEL_4A@1** | 0.485 (PE 71) | 7.70 | 20.268 (PE 123) | 19.78 |
| **EG_CORRECT_TRACERS_UKCA@1** | 7.237 (PE 120) | 7.62 | 7.839 (PE 48) | 0.60 |
| **GLUE_CONV_6A@1** | 3.389 (PE 101) | 6.68 | 12.87 (PE 59) | 9.48 |
| **GLUE_CONV_6A@2** | 3.35 (PE 65) | 6.62 | 12.605 (PE 77) | 9.26 |
| **UM_WRITDUMP@1** | 0.473 (PE 0) | 6.32 | 6.632 (PE 2) | 6.16 |
| **EG_INTERPOLATION_ETA@1** | 4.433 (PE 55) | 6.29 | 10.008 (PE 117) | 5.57 |
| **HALO_EXCHANGE:SWAP_BOUNDS_EW_DP@1** | 5.804 (PE 13) | 6.20 | 6.645 (PE 124) | 0.84 |

# Comparison of memory layout for segment method

UNIVERSITY OF LEEDS



# Code restructure for segment of columns

UNIVERSITY OF LEEDS

## Original

```
nbox = ni*nk*nl
t1d = reshape(t3d, nbox)
…
call aero_step(t1d,nbox)
…
mode_tracers = reshape(ae_nd, nk,ni,nl)
…
! mode_tracers returned to atmosphere code
```

## Modified as in vn8.6

```
DO ii = 1,  rows
  DO ik = 1, nseg
     ! Extract columns for this chunk
     jl =0
     DO kk = k_lo, k_up
        DO L = 1, model_levels
          jl = jl +1
          t1d(jl) = t3d(kk,ii,l)
        END DO
     END DO
     nbox_chunk = jl
     call aero_step(t1d, nbox_chunk
     jl =0
     DO kk = k_lo, k_up
       DO L = 1, model_levels
         jl = jl +1
         mode_tracers(kk,ii,l) = ae_nd(jl)
       END DO
     END DO
  END DO
END DO
! Mode_tracers returned to atmosphere code
```

## Modified for cache blocking vn10.5

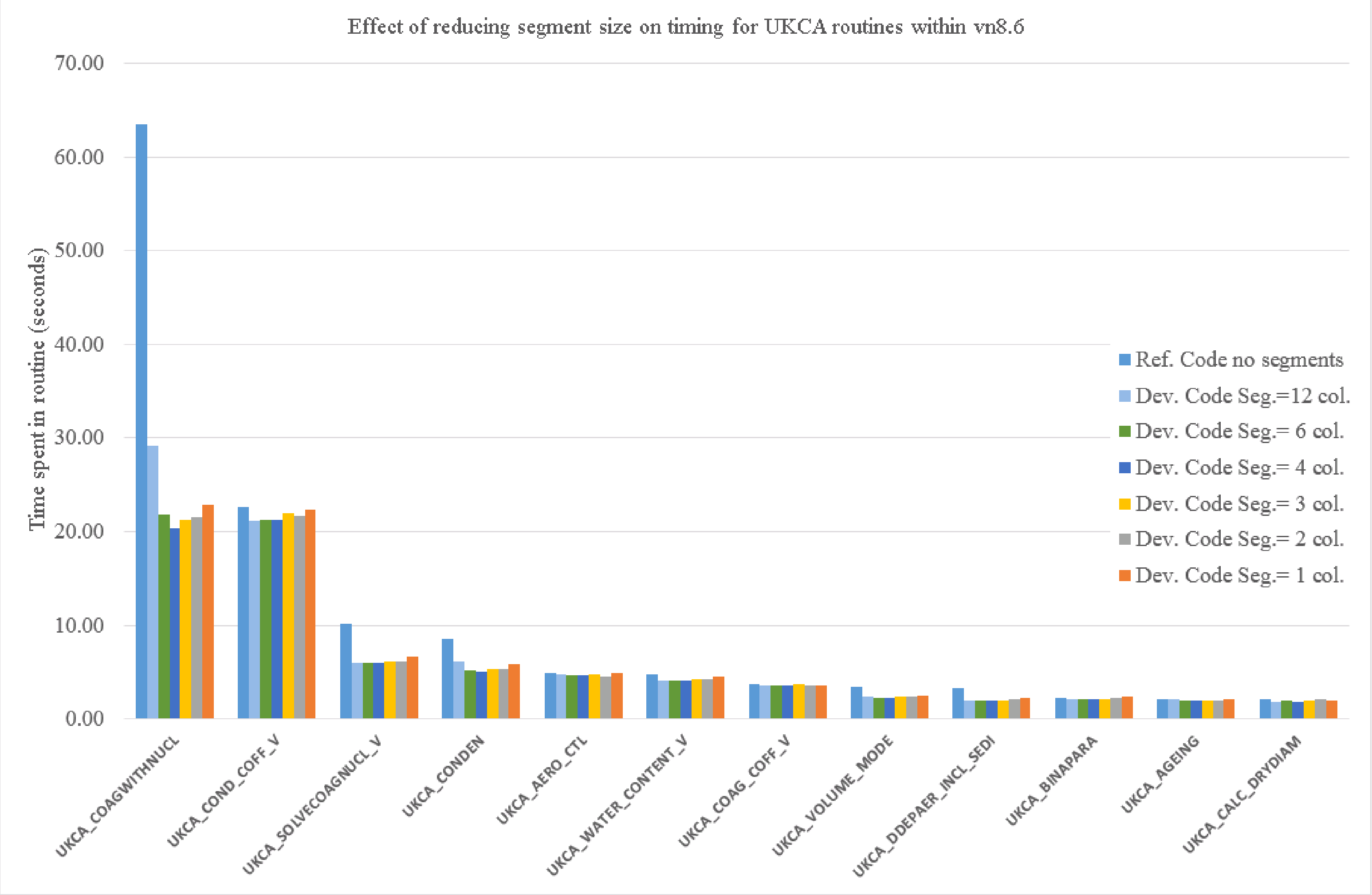
```
DO ik = 1,  nseg
  ! Extract columns for this seg
  CALL extract_segment(t3d,t1d)

  CALL aero_step(t1d, nbox_seg,ae
  ! Mode_tracers returned to atm
  CALL insert_segment(ae_nd, mod
END DO
```

# Effect of reducing the segment size, no OpenMP

UNIVERSITY OF LEEDS

Version 8.6 (In May 2016) at Cray User Group Meeting, London

- However, latest implementation in **vn10.5** shows no significant improvement because there was an interim “fix” applying chunking and OpenMP within UKCA_AERO_STEP()
- Achieved with arbitrary but informed knowledge and experience of HPC team at UKMO.
- Will have to rerun cases with that undone to show genuine effect of segment method as data blocking for cache



# Chasing release schedule

UNIVERSITY OF LEEDS

- Might be pioneering an actual “academic” contributing directly to lodging code
- Had to learn procedure for lodging code
- Competing with 5 or more code developments within same section of code
- When a new version is released
    - Make a  new branch
    - Merge in my changes (often difficult due to number of changes lodged by other developers)
    - Re-test with my development case
- Then do following
    - Rose stem (more than 100 tests to ensure no side effects from my changes)
    - Documentation
    - Sci/tech review
    - Code review
- I missed; vn10.2, vn 10.4 and vn10.5
    - Mainly due to decision to enhance before next deadline
- Got segment method into vn10.6 (hoorah!)

# Add Open MP parallelism

UNIVERSITY OF LEEDS

## Original

```
nbox = ni*nk*nl
t1d=reshape(t3d, nbox)
call aero_step(t1d,nbox,ae_nd)
mode_tracers=reshape(ae_nd, nk,ni,nl)
! mode_tracers returned to atmosphere
```

## Modified for cache blocking

```
DO ik = 1,  nseg
  ! Extract columns for this segment
  CALL extract_segment(t3d,t1d)

  CALL aero_step(t1d, nbox_seg,ae_nd)

  ! Mode_tracers returned to atmosphere code
  CALL insert_segment(ae_nd, mode_tracers)
END DO
```

Significant effort in code rewriting to make the gains described here.

## Modified for OpenMP

```
!$OMP PARALLEL DEFAULT(NONE) SHARED(…) PRIVATE(…)
<some preliminary work>
ALLOCATE(t1d,nbmax) ! Assume uniform
ALLOCATE(ae_nd,nbmax,nmode) !2D
!$OMP DO SCHEDULE(STATIC)
DO ik = 1,  nseg
  nbs=nbox_seg(ik)
  IF(nonuniform_seg) THEN
     IF(ALLOCATED(ae_nd) THEN DEALLOCATE(ae_nd)
     ALLOCATE(ae_nd,nbs,nmode)
  END IF
  ! Extract columns for this segment
  CALL extract_segment(t3d,t1d)

  CALL aero_step(t1d, nbs,ae_nd)

  ! Mode_tracers returned to atmosphere code
  CALL insert_segment(ae_nd, mode_tracers)
END DO
!$OMP END DO
<some work round up>
CALL DEALLOCATE(ae_nd)
!$OMP END PARALLEL
! Mode_tracers returned to atmosphere
```

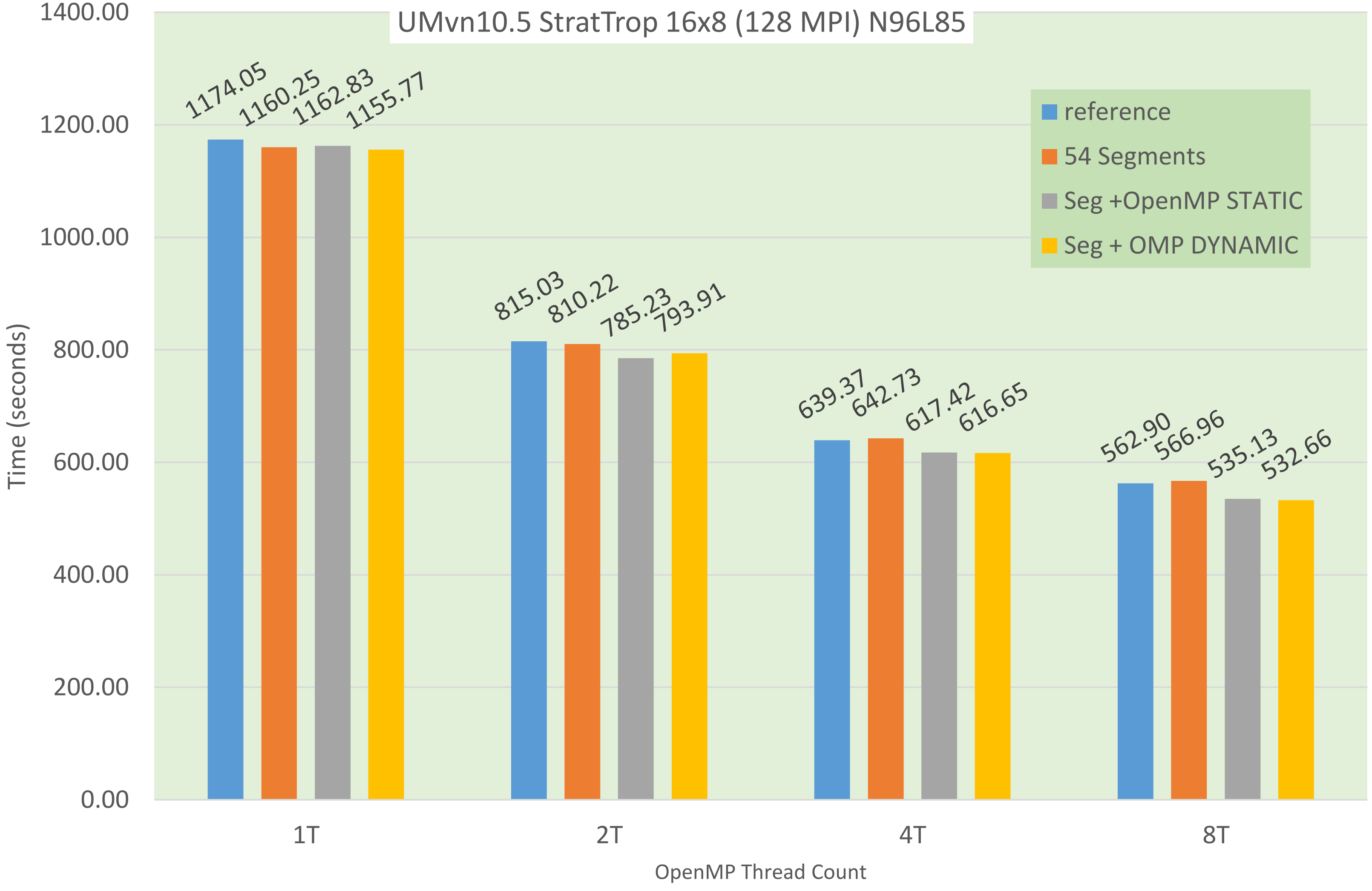
# StraTrop 16x8 Effect of OpenMP

UNIVERSITY OF LEEDS

- UM vn10.5
- StraTrop 16x8 N96L85 (2day)
- 144 atmosphere time steps (20min)

- 3% improvement over 2T
- 5% improvement possible

- There is active chemistry processing not yet withinOpenMP



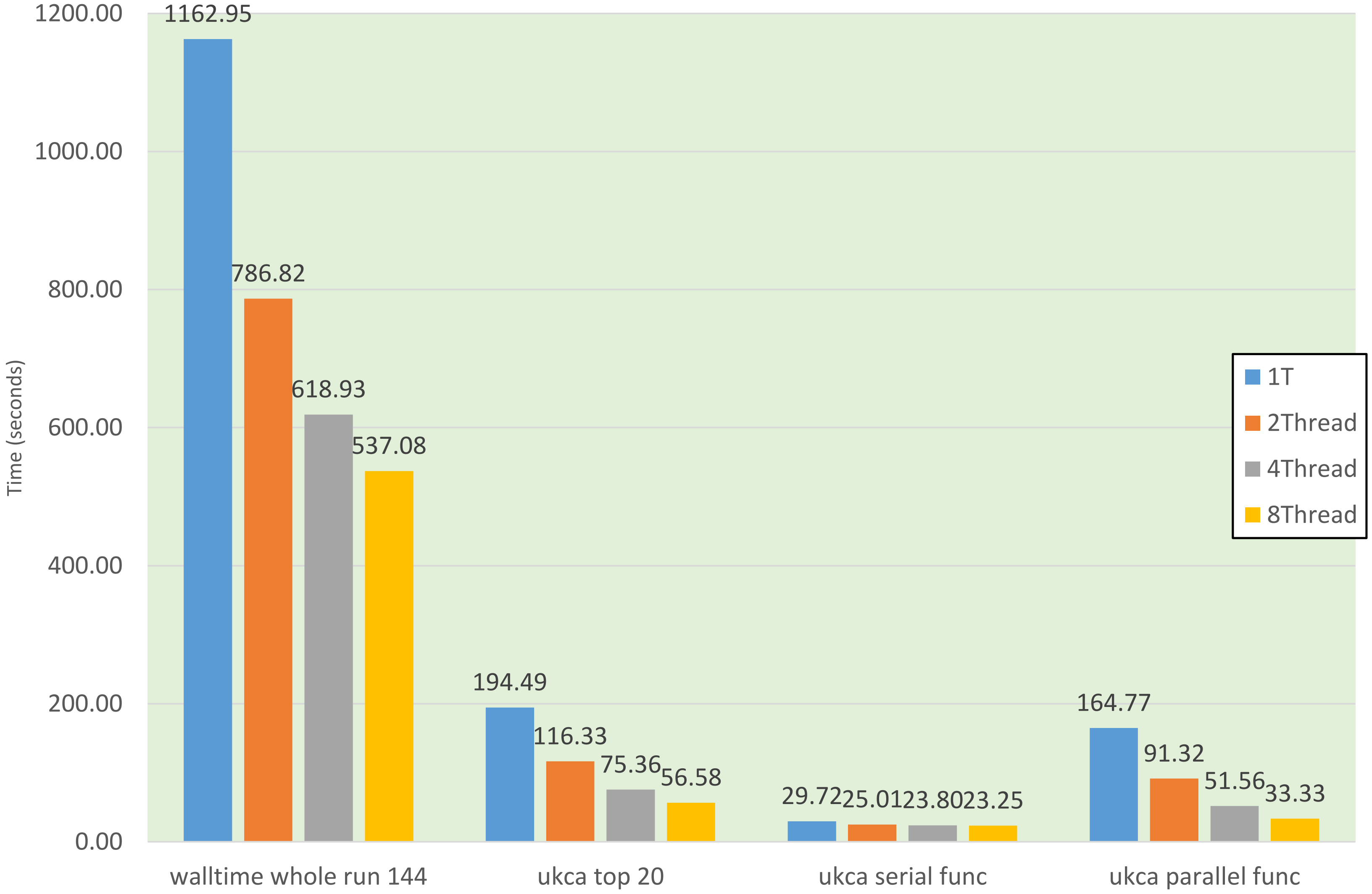
# Break down of OpenMP in UKCA

UNIVERSITY OF LEEDS

UM vn10.5 + 54 Segments
StratTrop 16x8 N96L85 (2day)
144 atmosphere time steps (20min)

Chart shows the time for top 20 UKCA functions

Then separated into serial and parallel (some UKCA outside OpenMP parallel region)



# Improvement for UKCA compared to whole simulation

UNIVERSITY OF LEEDS

- The percentage of the runtime that is spent in UKCA has been reduced
- Reducing perceived “overhead”

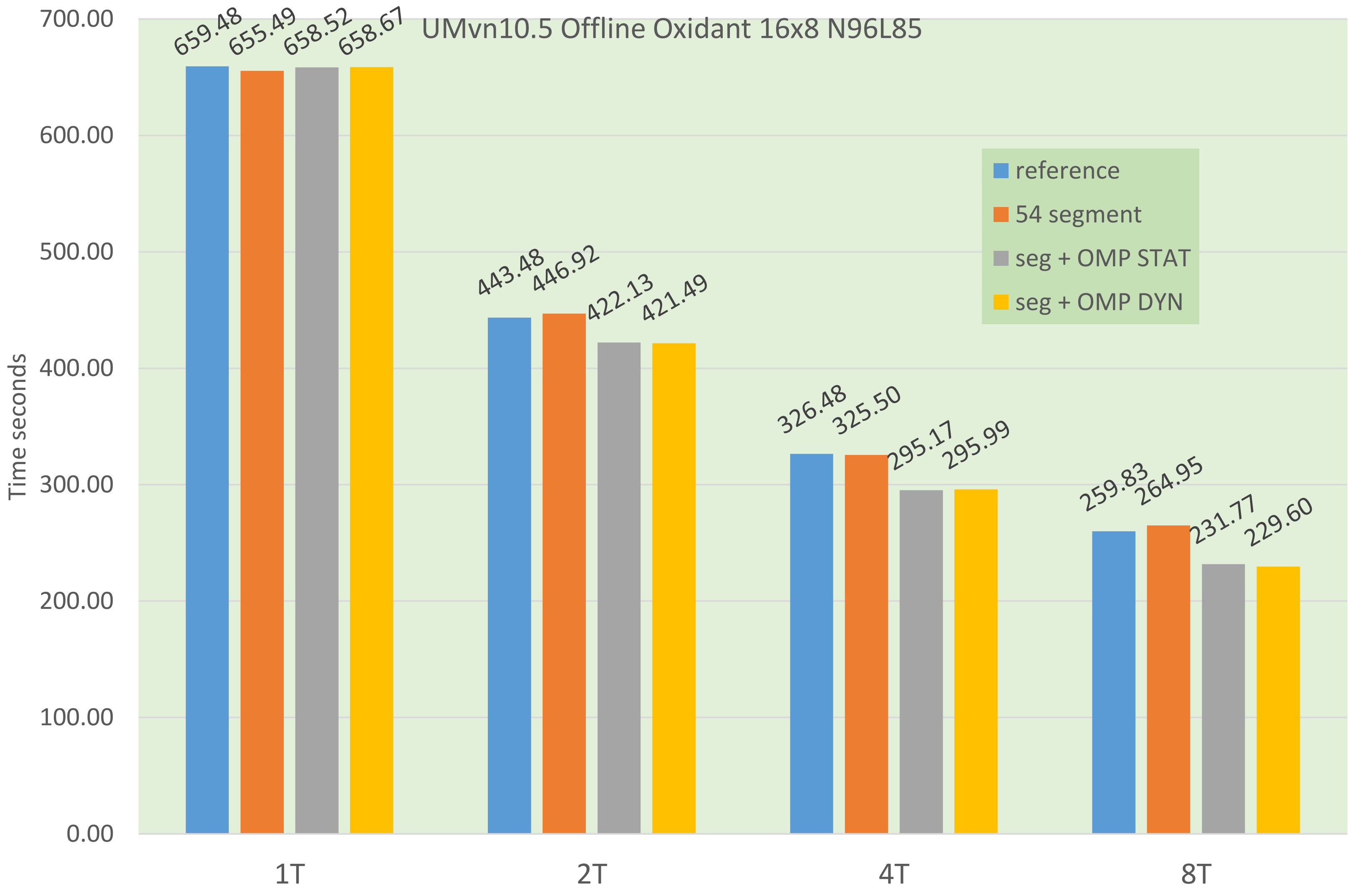
| strattrop16x8 | 1Thread | 2Thread | 4Thread | 8Thread |
|---|---|---|---|---|
| walltime whole run 144 | 1162.95 | 786.82 | 618.93 | 537.08 |
| **ukca top 20** | **194.49** | **116.33** | **75.36** | **56.58** |
| ukca serial func | 29.72 | 25.01 | 23.80 | 23.25 |
| ukca parallel func | 164.77 | 91.32 | 51.56 | 33.33 |
| percent ukca20 | 0.17 | 0.15 | 0.12 | 0.11 |
| percent ukca20 serial | 0.03 | 0.03 | 0.04 | 0.04 |
| percent ukca20 parallel | 0.14 | 0.12 | 0.08 | 0.06 |

# The Offline Oxidant 4 configurations

UNIVERSITY OF LEEDS

- UM vn10.5
- OfflineOx 16x8 N96L85 (2day)
- 144 atmosphere time steps (20min)

- 5% improvement over 2T
- 10% improvement possible
- Require 4x resource



# Comparing OpenMP Schedules: Static and Dynamic

UNIVERSITY OF LEEDS

- Choosing uniform segments was important when no OpenMP
  - First version in vn8.6 had a restriction to row length maximum segment size
  - Moved on since (MAY) can now be flexible in segment size
- STATIC distributes rows evenly
  - Noticed that even when 8 threads assigned the scheduler chose only 6
  - Additional difficulty in prescribing 6 threads specifically without intervention
- DYNAMIC will allocate all threads some of the work (iterations)
  - Remaining iterations are allocating as threads become available
  - Only tested default chunk size (OMP chooses)
  - Might see variation if we allocate groups of segments to different threads

# Summary

UNIVERSITY OF LEEDS

- Currently status
  - Early development work within vn8.6 completed
  - Restructured code to send smaller amount of data to GLOMAP
  - Reduce the time spent in GLOMAP
  - Overall runtime reduction by 3-10%
  - Activated OpenMP around GLOMAP within a vn10.4 branch
  - **Added the segmentation method to vn10.5 – NOW COMMMITTED officially in vn10.6!**
  - Added OpenMP to another vn10.5 development branch
- Ongoing plan
  - Migrate vn10.5 OpenMP development into a vn10.6 branch
  - Start work on a case that has active chemistry

# Current role: Centre for Excellence in Modelling Atmosphere and Climate

UNIVERSITY OF LEEDS

- The Centre of Excellence for Modelling the Atmosphere and Climate (CEMAC)
  - is a major new initiative within the Institute for Climate and Atmospheric Science (ICAS) and School of Earth & Environment (SEE) at the University of Leeds.
  - CEMAC aims to be the UK's leading centre of excellence in atmospheric & climate modelling and complex data exploitation. Its vision is to significantly enhance and accelerate our high impact research in weather, climate and atmospheric composition, and to train and educate a new generation of students and scientists in the latest techniques in scientific computing and data processing & visualisation.
- ICAS has nearly 200 researchers (Ph.D. Student to Professor levels)
  - 20 Lead researchers
  - 3 research areas
    - Climate change
    - Atmospheric and Cloud dynamics
    - Atmospheric Chemistry and Aerosol processes