Porting and Optimisation of UM on ARCHER

Karthee Sivalingam, NCAS-CMS

HPC Workshop ECMWF

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Met Office

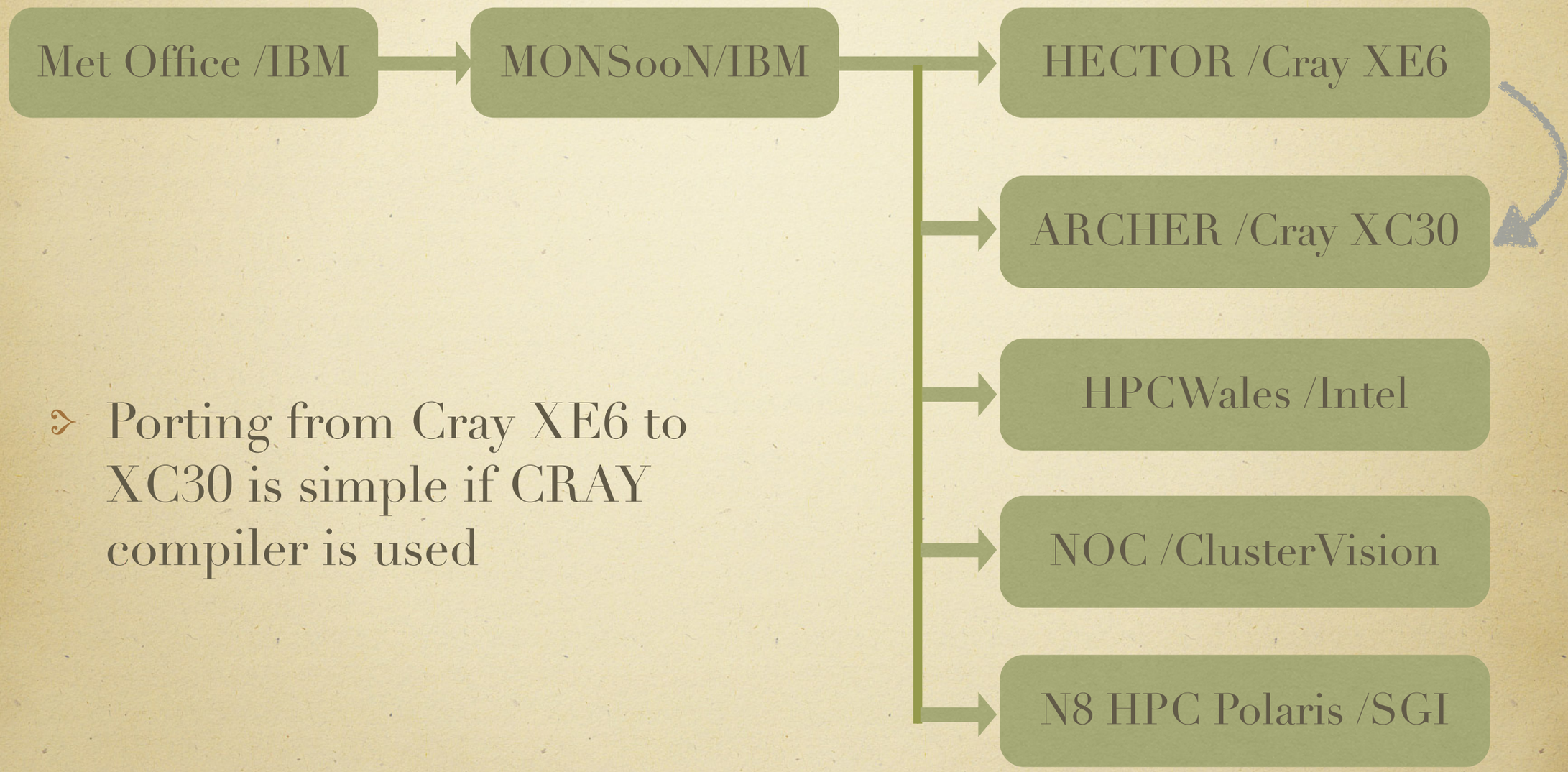
Acknowledgements

- **NCAS-CMS**
  - Bryan Lawrence
  - Jeffrey Cole
  - Rosalyn Hatcher
  - Andrew Heaps
  - David Hassell
  - Grenville Lister
  - William McGinty
  - Simon Wilson
  - Annette Osprey
- **Met Office**
  - Mick Carter
  - Jean-Christophe Rioual
  - Matthew Hambley
  - Jamie Kettleborough
- **CRAY & EPCC**

Summary

- Porting UM to ARCHER (XC30)
- Initial Performance on ARCHER
- Performance analysis
- Tuning MPI ~~and IO~~
- Optimising thread performance

Porting



- Porting from Cray XE6 to XC30 is simple if CRAY compiler is used

HECTOR and ARCHER

	AMD Opteron	Intel Xeon
Clock speed	2.3	2.7
Cores per node	32	24
threads per core	1	2
vector	sse	avx
bits wide	128	256
L2 cache	512 KB	256 KB
L3 cache	6 MB	30 MB
Peak DP flops per node	294 GFlops	518 GFlops

Memory bandwidth , MPI bandwidth and IO bandwidth ?

ARCHER …

- 3008 compute nodes
- Dual socket , 12 cores - forming 2 NUMA nodes
- 64 (or 128) GB memory per node
- Cray Aries network
- Cray XC30 Dragonfly Topology
- Copper and Optical cabling
- Lustre parallel file system

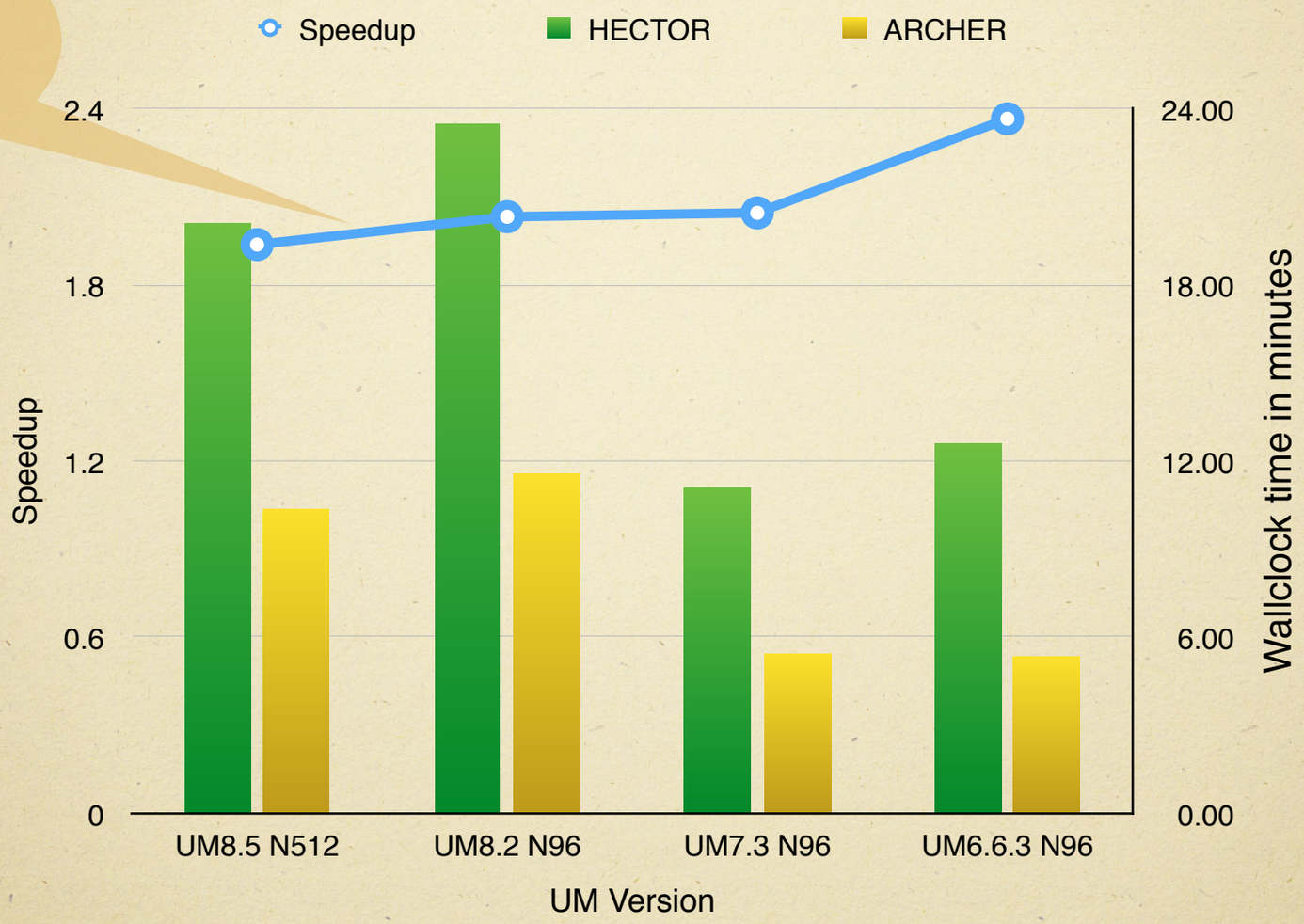
UM jobs …

Job name	Columns	Rows	Land points	Vertical levels
N96	192	144	11271	85
N216	432	324	52614	85
N512	1024	768	280592	85

All the jobs are standard jobs ported from MetOffice to ARCHER. All the jobs use Cray compiler and strict bit comparability is enforced.

Initial Performance Comparison

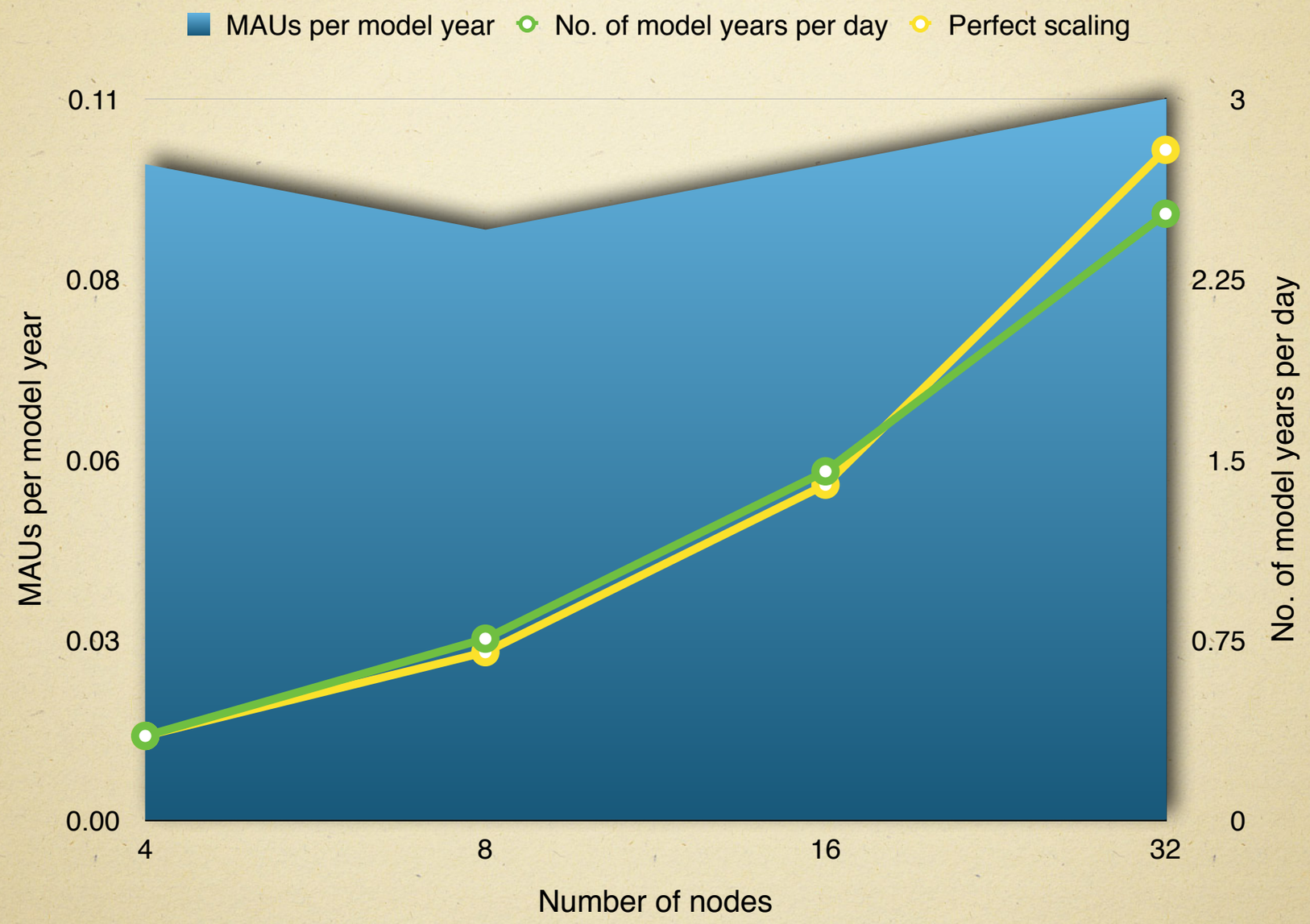
4 times throughput?



Above jobs in HECTOR and ARCHER use similar MPI decomposition and same number of threads

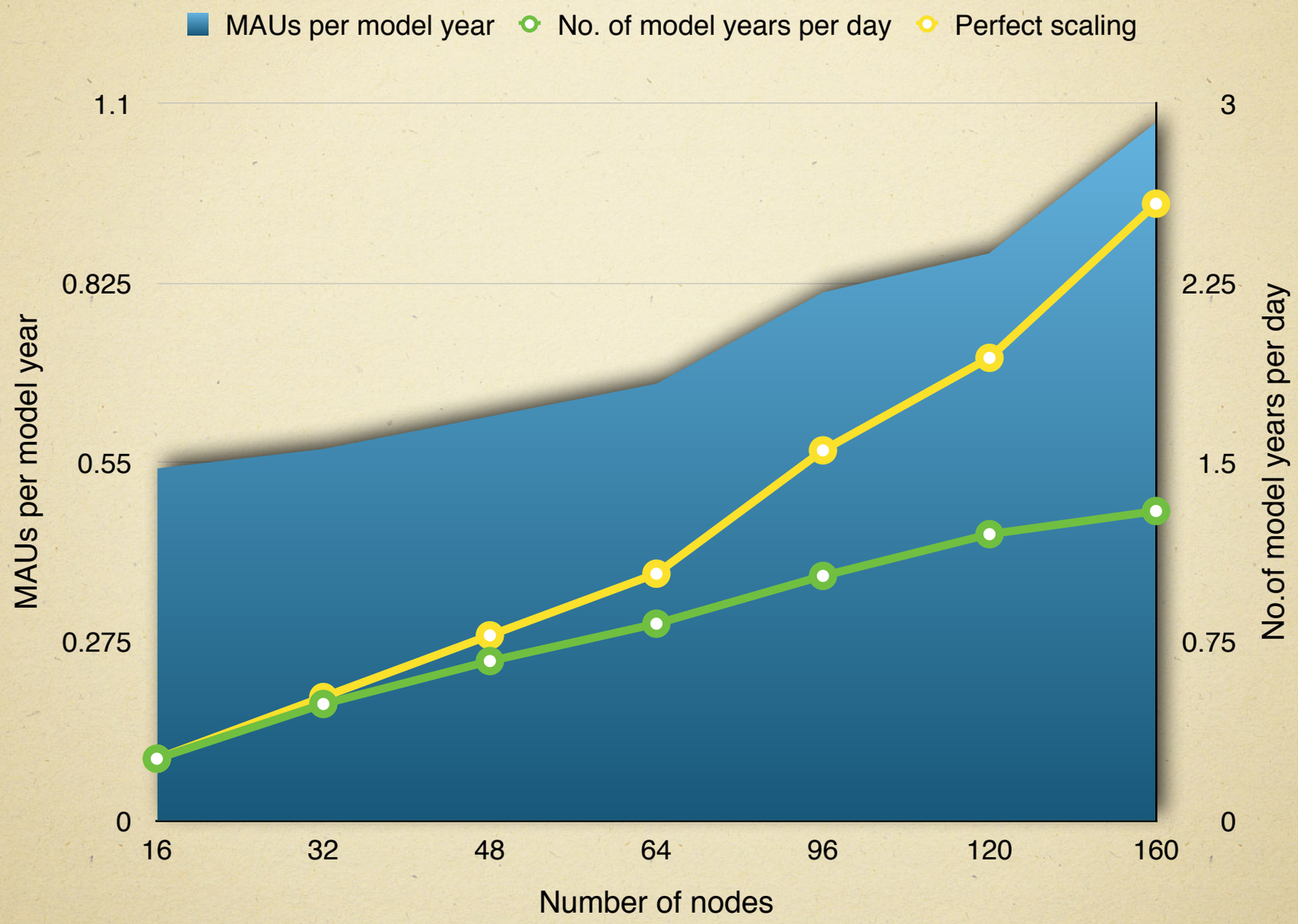
Wallclock time refers to the time taken to complete simulation of 1 model day

N96 job on ARCHER



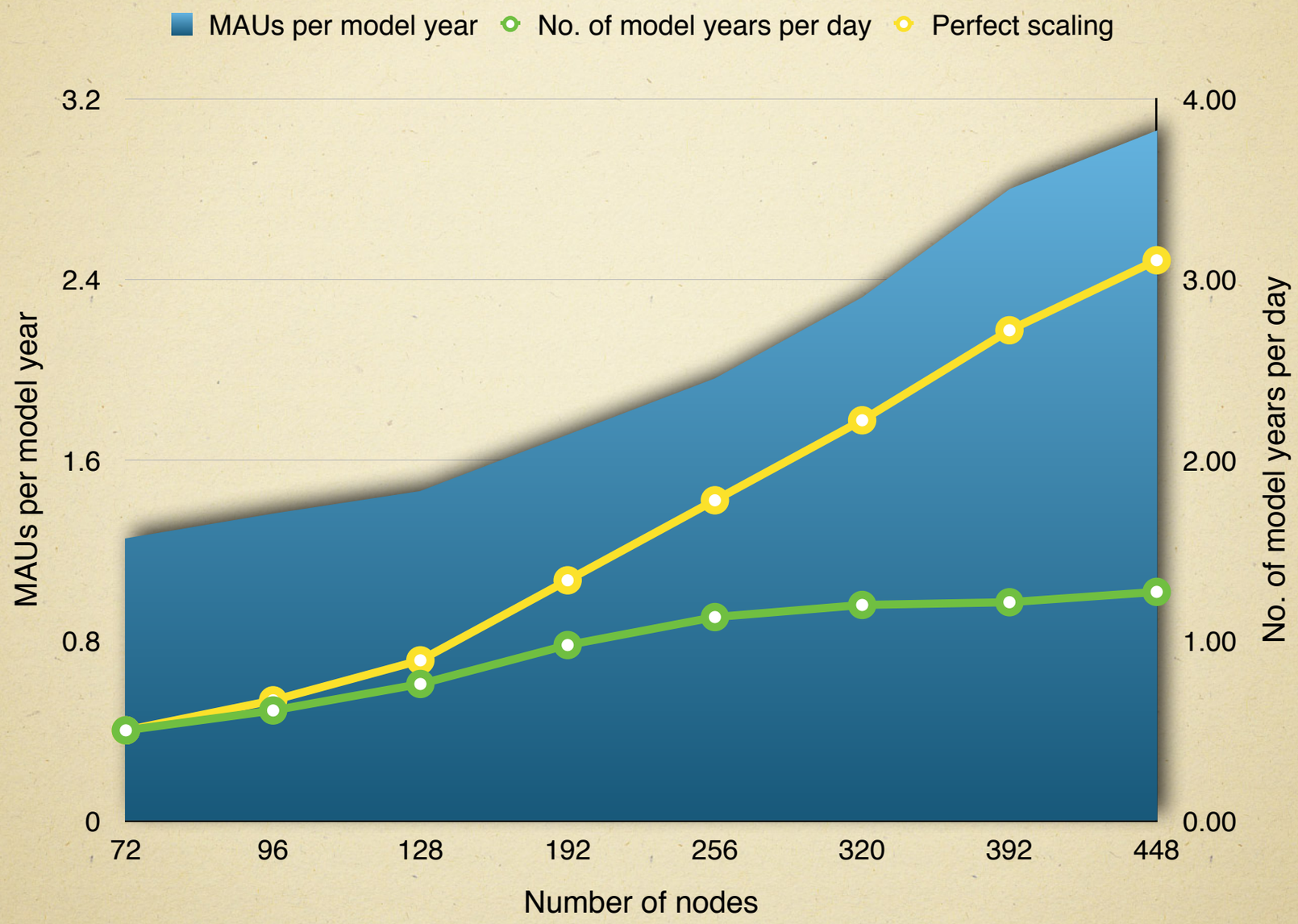
All the above runs uses 12 MPI PEs and 2 threads per node. Cost per MAU of ARCHER is £1640

N216 job on ARCHER



All the above runs uses 12 MPI PEs and 2 threads per node. Cost per MAU of ARCHER is £1640.

N512 job on ARCHER



All the above runs uses 12 MPI PEs and 2 threads per node. Cost per MAU of ARCHER is £1640.

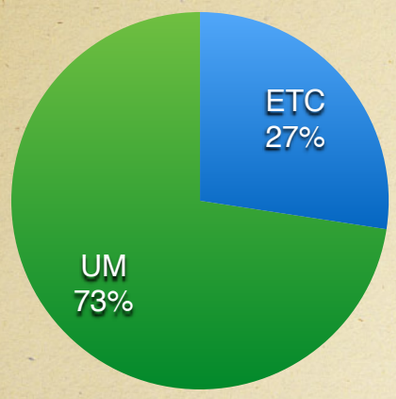
When number of PEs in EW or NS goes more that 120 or 60 respectively, the model does not bit compare.

What to do?

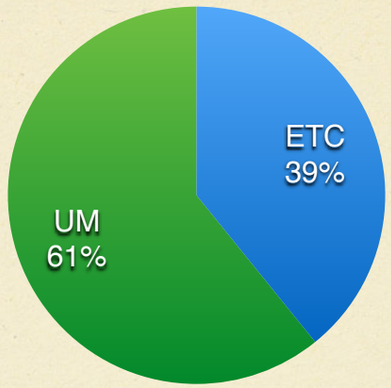
- N512 job does not scale beyond 250 nodes.
- N216 job does not scale beyond 96 nodes.
- Is 12 MPI PEs per node ideal? Is 2 threads per PE ideal ? Can we use hardware threads ?
- Need to analyse the performance.
- CRAY performance analysis tools readily available.
  - CrayPAT
  - Apprentice

Profile analysis

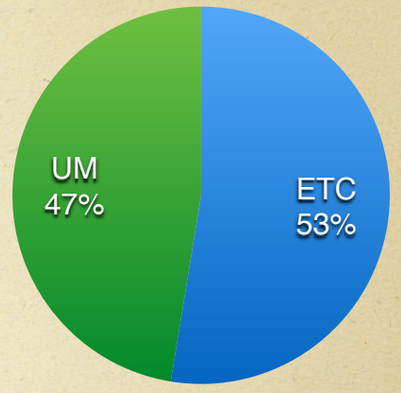
N96 - 8 nodes

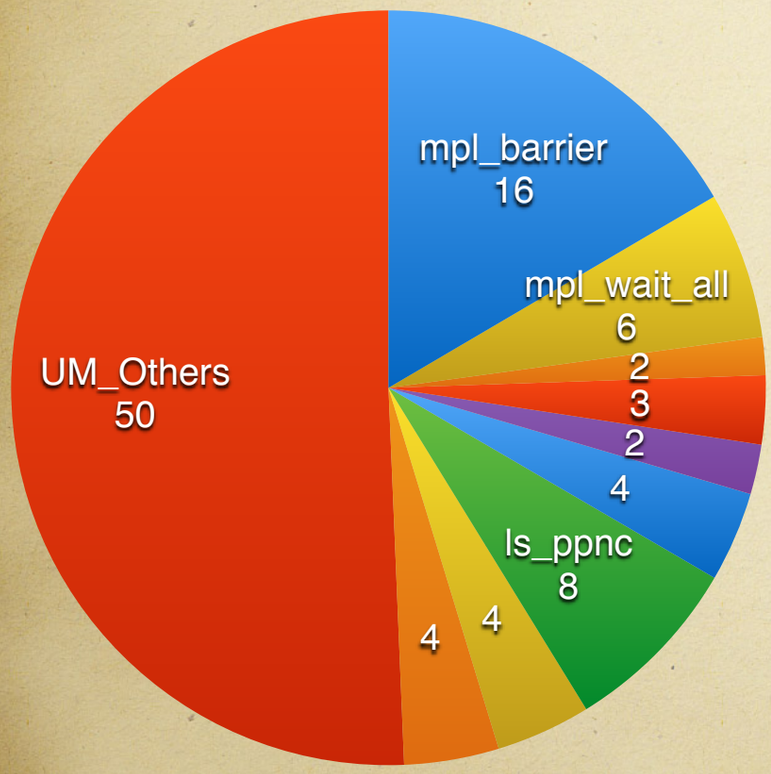


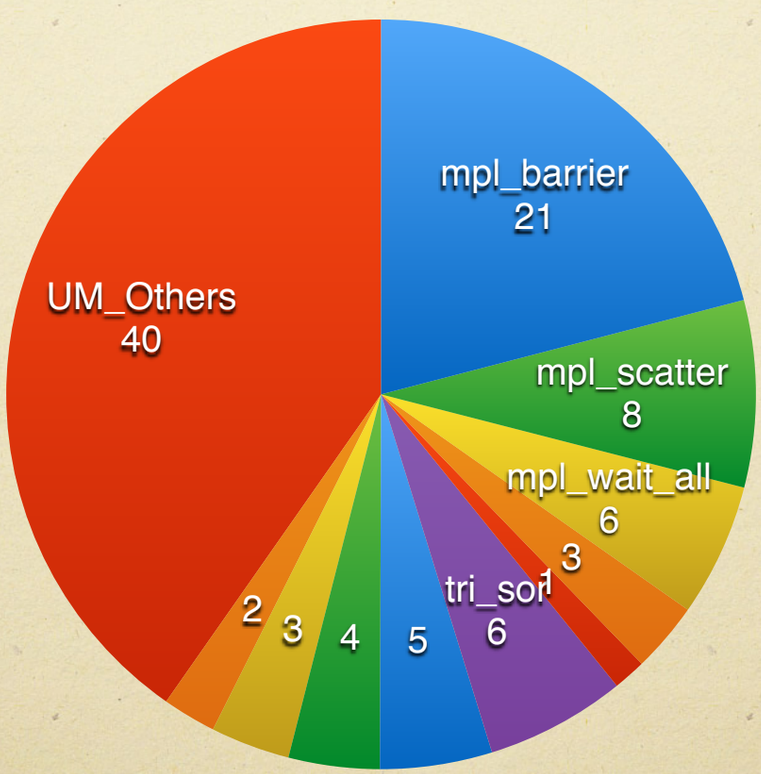
N512 - 72 nodes

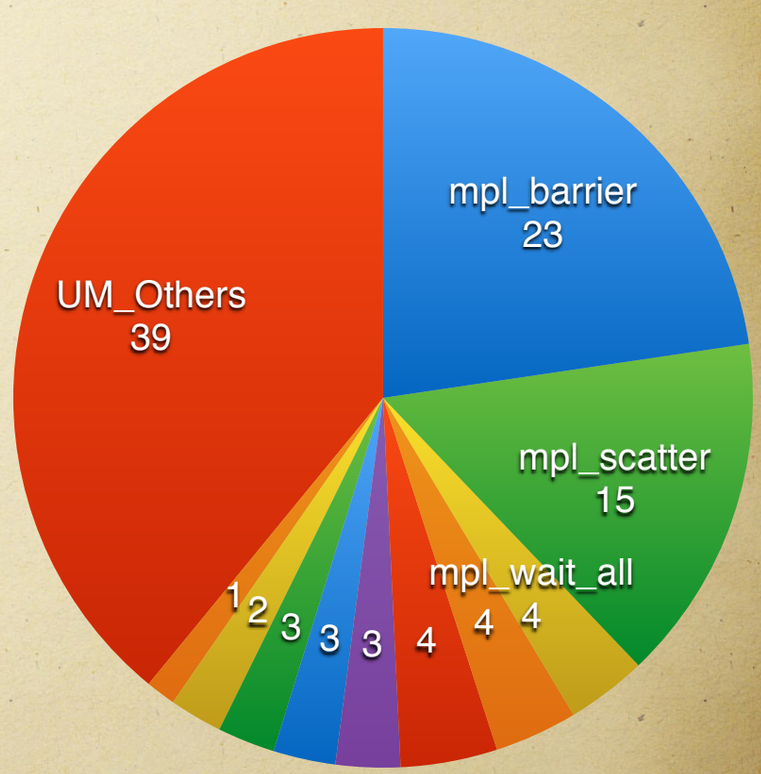


N512 - 192 nodes









mpl_barrier
tri_sor
UM_Others
mpl_scatter
eg_cubic_lagrange
mpl_wait_all
ls_ppnc
mpl_allreduce
ni_conv_ctl
ETC_Others
glue_rad

MPI Bandwidth

- Message passing in UM is done only by thread 0.
- Message passing overhead increases by 14% as the number of nodes is increased from 72 to 192.
- Scaling of MPI bandwidth improved by using
  - MPI Asynchronous progress
  - Use unassigned CPUS for MPI
  - MPI rank placement (Grid order)

PE decomposition 24 x 36

0	1	2	3	4	…	35
36	37	38	39	40	…	71
72	73	74	75	76	…	107
108	109	110	111	112	…	143
…	…	…	…	…	…	…
828	829	829	829	829	…	863

Nearest neighbour communications

12 PEs per node

SMP - MPI Rank Order

Node 0

0	7	8
1	6	9
2	5	10
3	4	11

Node 4

36	43	44
37	42	45
38	41	46
39	40	47

Node 10

108	115	116
109	114	117
110	113	118
111	112	119

Node 6

72	79	80
73	78	81
74	77	82
75	76	83

GRID - MPI Rank Order

Node 0

0	1	2
36	37	38
72	73	74
108	109	110

Node 4

111	112	113
75	76	77
39	40	41
3	4	5

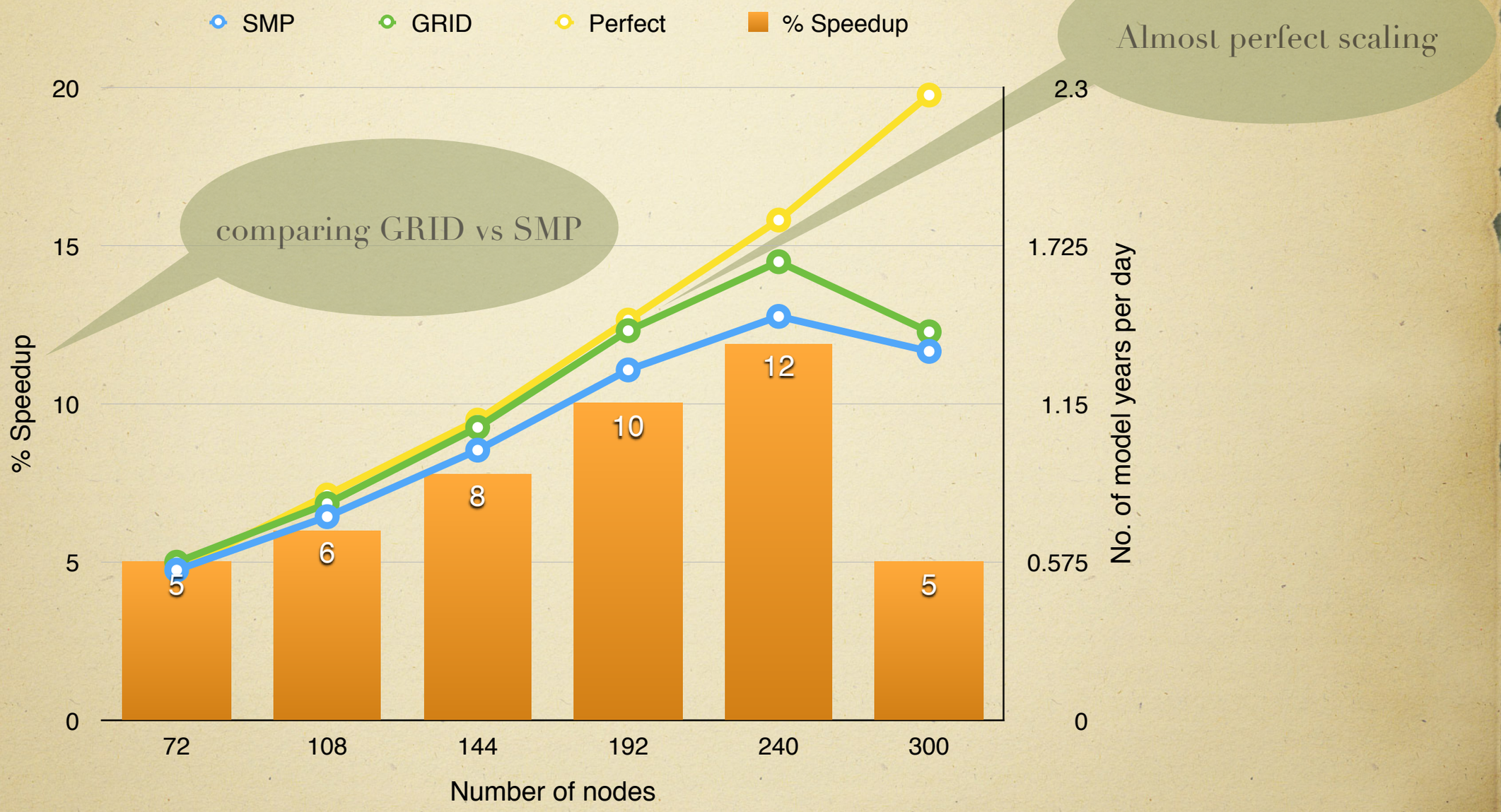
Node 2

144	145	146
180	181	182
216	217	218
252	253	254

Node 3

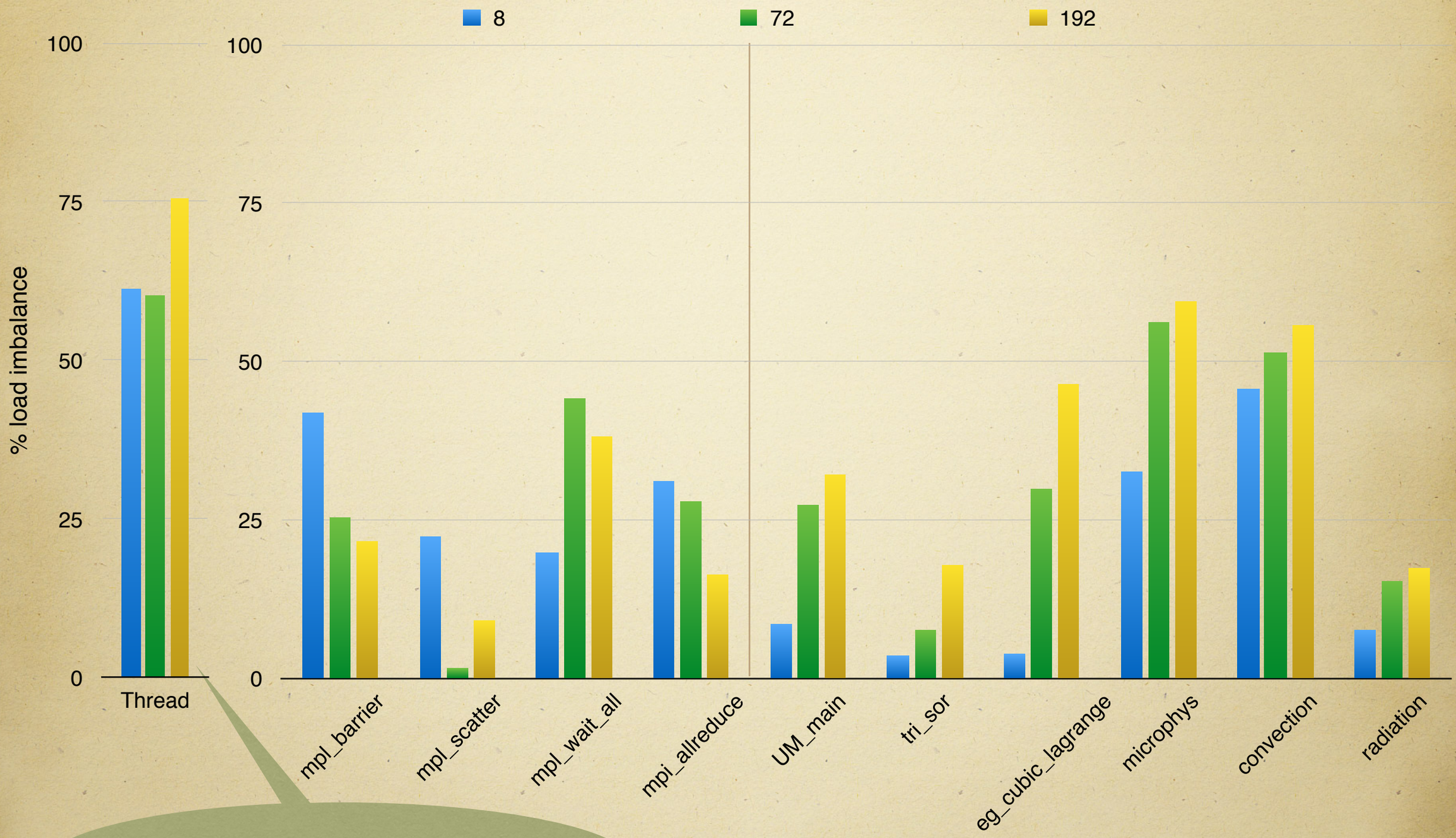
255	256	257
219	220	221
183	184	185
147	148	149

GRID vs SMP MPI rank order



All the above results are obtained using 12 PEs per node and two OPENMP threads. IO is not included in the measurement as the IO resources are shared and accurate measurement is difficult.

Load imbalance



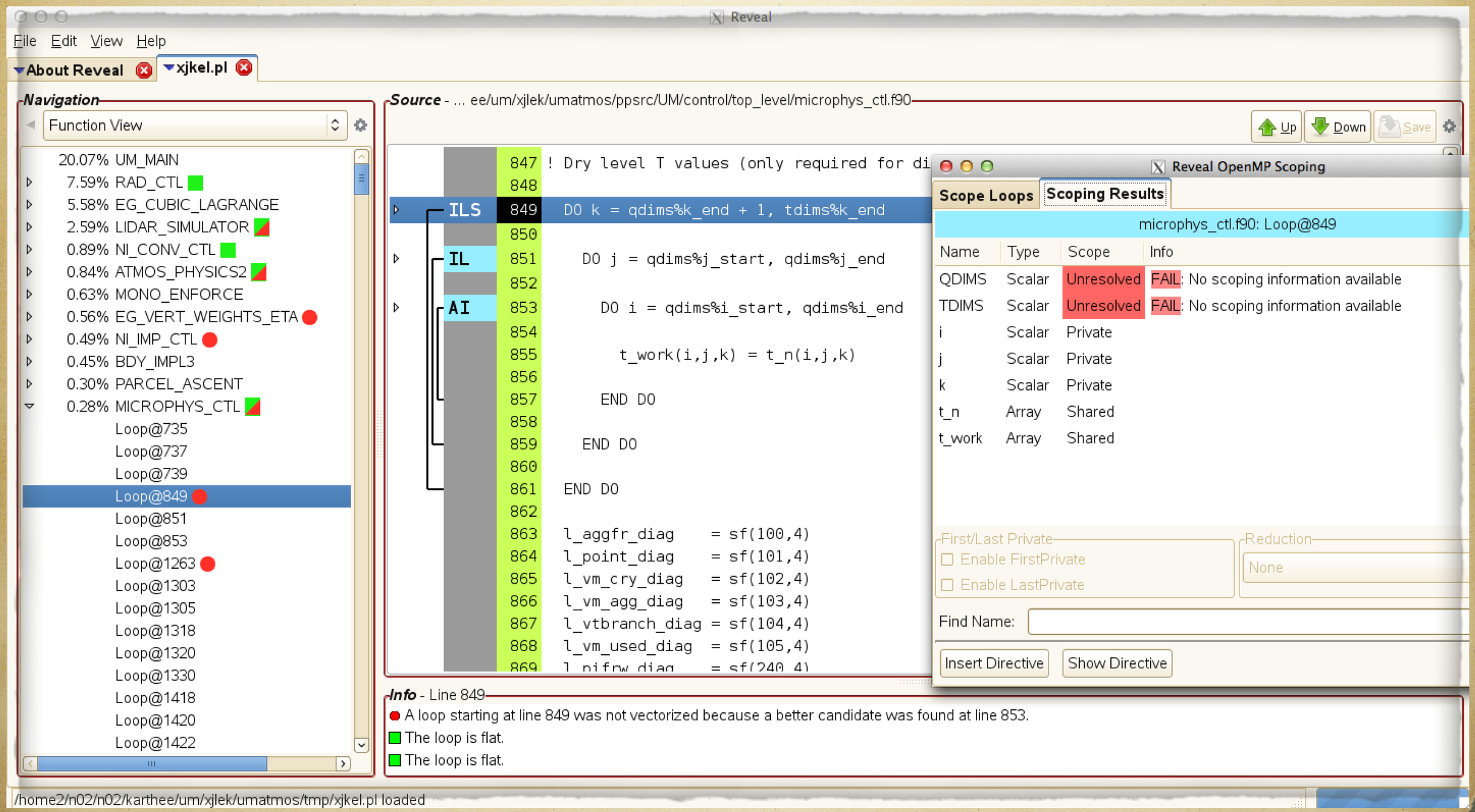
Can expect poor thread scaling

OPENMP directives

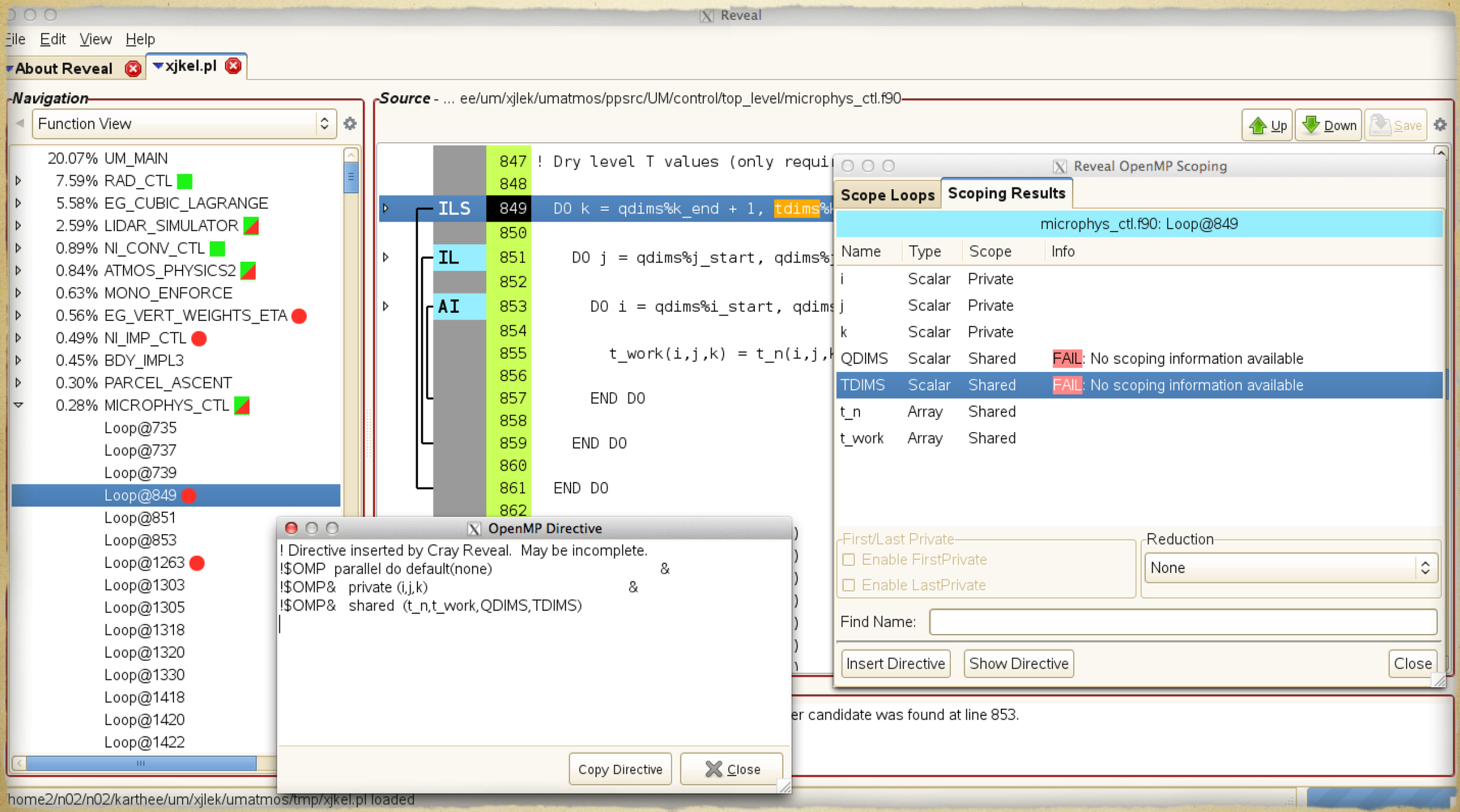
- 60% to 75% thread imbalance in UM jobs.
- coverage of OPENMP regions is less than 50%.
- Cray Reveal - integrated performance analysis and code optimisation tool.
    - provides loop analysis and scoping of serial loops.
    - suggests OPENMP directive that can be inserted to a loop.
    - can attach the performance data collected during execution to identify profile of loops.
    - requires knowledge of OPENMP to resolve conflicts and issues.
    - works only with Cray compiling environment.
    - does not support tasks, barrier, critical or atomic regions.

For more details - refer Cray documentation (not much)

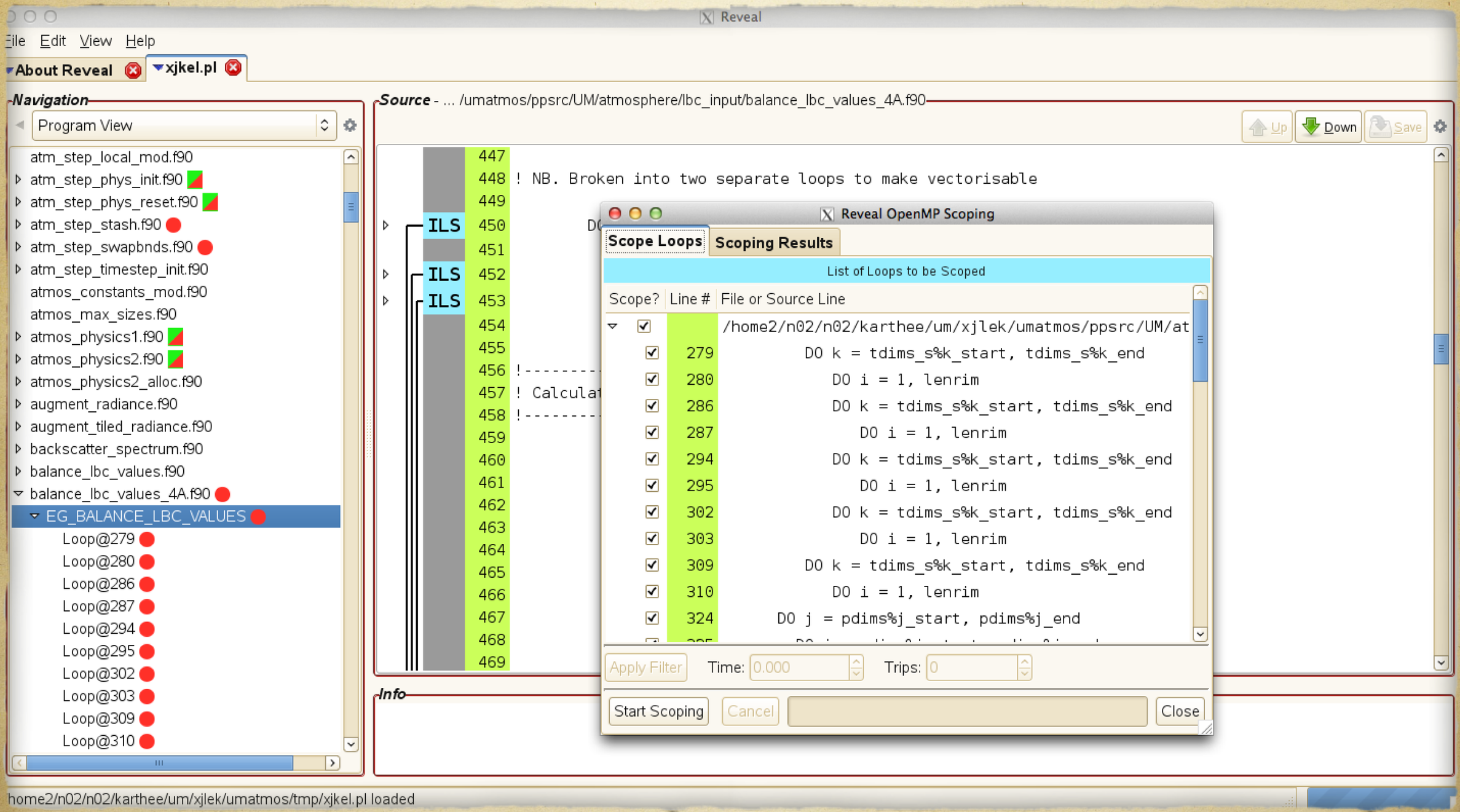
CRAY Reveal


Reveal
File Edit View Help
About Reveal
xjkel.pl
Navigation
Function View
20.07% UM_MAIN
7.59% RAD_CTL
5.58% EG_CUBIC_LAGRANGE
2.59% LIDAR_SIMULATOR
0.89% NI_CONV_CTL
0.84% ATMOS_PHYSICS2
0.63% MONO_ENFORCE
0.56% EG_VERT_WEIGHTS_ETA
0.49% NI_IMP_CTL
0.45% BDY_IMPL3
0.30% PARCEL_ASCENT
0.28% MICROPHYS_CTL
Loop@735
Loop@737
Loop@739
Loop@849
Loop@851
Loop@853
Loop@1263
Loop@1303
Loop@1305
Loop@1318
Loop@1320
Loop@1330
Loop@1418
Loop@1420
Loop@1422
Source - ... ee/um/xjlek/umatmos/ppsrc/UM/control/top_level/microphys_ctl.f90
Up
Down
Save
847 ! Dry level T values (only required for di
848
ILS 849 DO k = qdims%k_end + 1, tdims%k_end
850
IL 851 DO j = qdims%j_start, qdims%j_end
852
AI 853 DO i = qdims%i_start, qdims%i_end
854
855 t_work(i,j,k) = t_n(i,j,k)
856
857 END DO
858
859 END DO
860
861 END DO
862
863 l_aggfr_diag = sf(100,4)
864 l_point_diag = sf(101,4)
865 l_vm_cry_diag = sf(102,4)
866 l_vm_agg_diag = sf(103,4)
867 l_vtbranch_diag = sf(104,4)
868 l_vm_used_diag = sf(105,4)
869 l_pifrw_diag = sf(240,4)
Reveal OpenMP Scoping
Scope Loops
Scoping Results
microphys_ctl.f90: Loop@849
Name Type Scope Info
QDIMS Scalar Unresolved FAIL: No scoping information available
TDIMS Scalar Unresolved FAIL: No scoping information available
i Scalar Private
j Scalar Private
k Scalar Private
t_n Array Shared
t_work Array Shared
First/Last Private
Enable FirstPrivate
Enable LastPrivate
Reduction
None
Find Name:
Insert Directive
Show Directive
Info - Line 849
A loop starting at line 849 was not vectorized because a better candidate was found at line 853.
The loop is flat.
The loop is flat.
/home2/n02/n02/karthee/um/xjlek/umatmos/tmp/xjkel.pl loaded

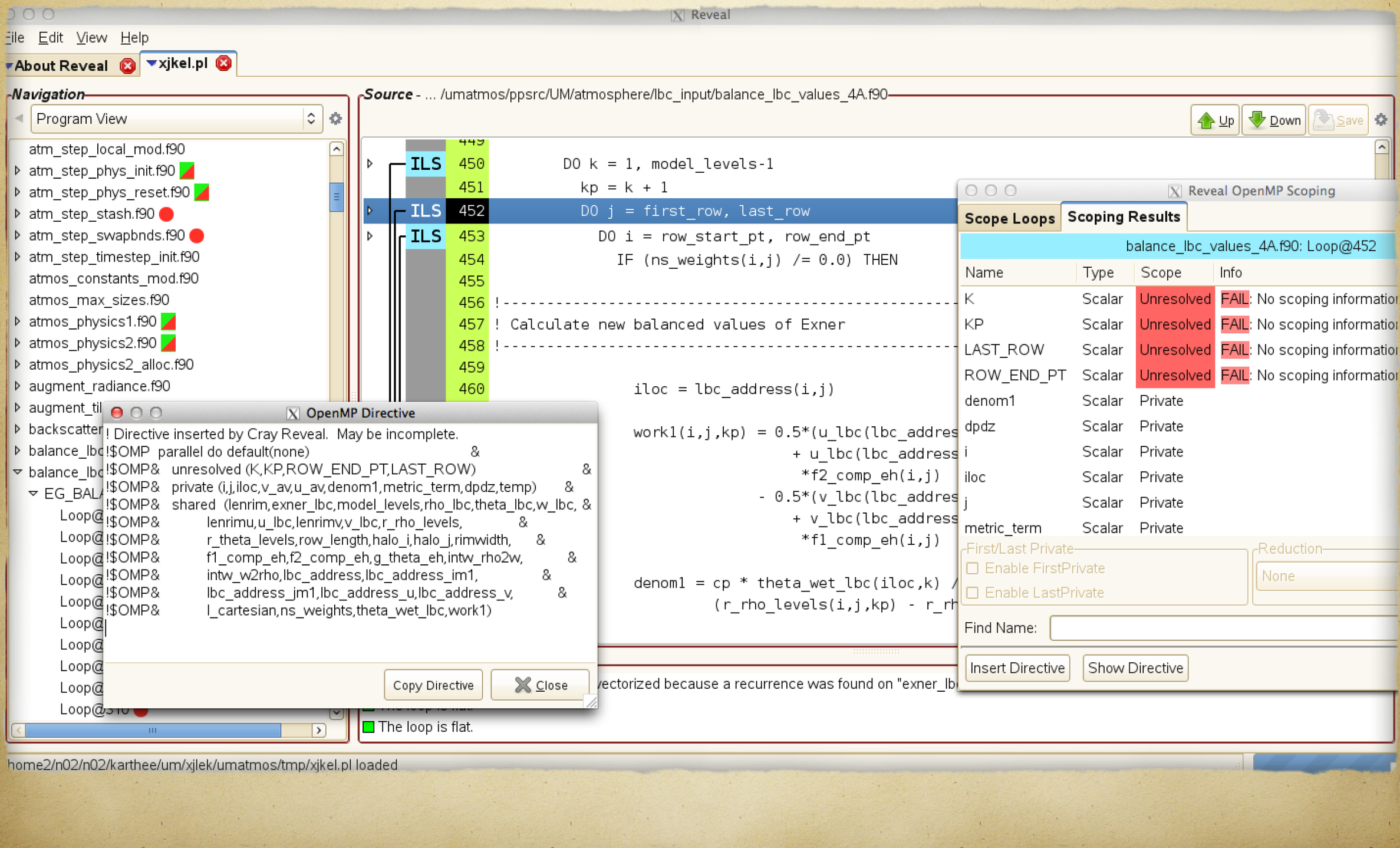
CRAY Reveal


Reveal
File Edit View Help
About Reveal
xjkel.pl
Navigation
Function View
20.07% UM_MAIN
7.59% RAD_CTL
5.58% EG_CUBIC_LAGRANGE
2.59% LIDAR_SIMULATOR
0.89% NI_CONV_CTL
0.84% ATMOS_PHYSICS2
0.63% MONO_ENFORCE
0.56% EG_VERT_WEIGHTS_ETA
0.49% NI_IMP_CTL
0.45% BDY_IMPL3
0.30% PARCEL_ASCENT
0.28% MICROPHYS_CTL
Loop@735
Loop@737
Loop@739
Loop@849
Loop@851
Loop@853
Loop@1263
Loop@1303
Loop@1305
Loop@1318
Loop@1320
Loop@1330
Loop@1418
Loop@1420
Loop@1422
Source - ... ee/um/xjlek/umatmos/ppsrc/UM/control/top_level/microphys_ctl.f90
Up
Down
Save
847 ! Dry level T values (only requi
848
ILS 849 DO k = qdims%k_end + 1, tdims%
850
IL 851 DO j = qdims%j_start, qdims%
852
AI 853 DO i = qdims%i_start, qdims
854
855 t_work(i,j,k) = t_n(i,j,k
856
857 END DO
858
859 END DO
860
861 END DO
862
Reveal OpenMP Scoping
Scope Loops
Scoping Results
microphys_ctl.f90: Loop@849
Name Type Scope Info
i Scalar Private
j Scalar Private
k Scalar Private
QDIMS Scalar Shared FAIL: No scoping information available
TDIMS Scalar Shared FAIL: No scoping information available
t_n Array Shared
t_work Array Shared
First/Last Private
Enable FirstPrivate
Enable LastPrivate
Reduction
None
Find Name:
Insert Directive
Show Directive
Close
OpenMP Directive
! Directive inserted by Cray Reveal. May be incomplete.
!$OMP parallel do default(none) &
!$OMP& private (i,j,k) &
!$OMP& shared (t_n,t_work,QDIMS,TDIMS)
Copy Directive
Close
er candidate was found at line 853.
home2/n02/n02/karthee/um/xjlek/umatmos/tmp/xjkel.pl loaded

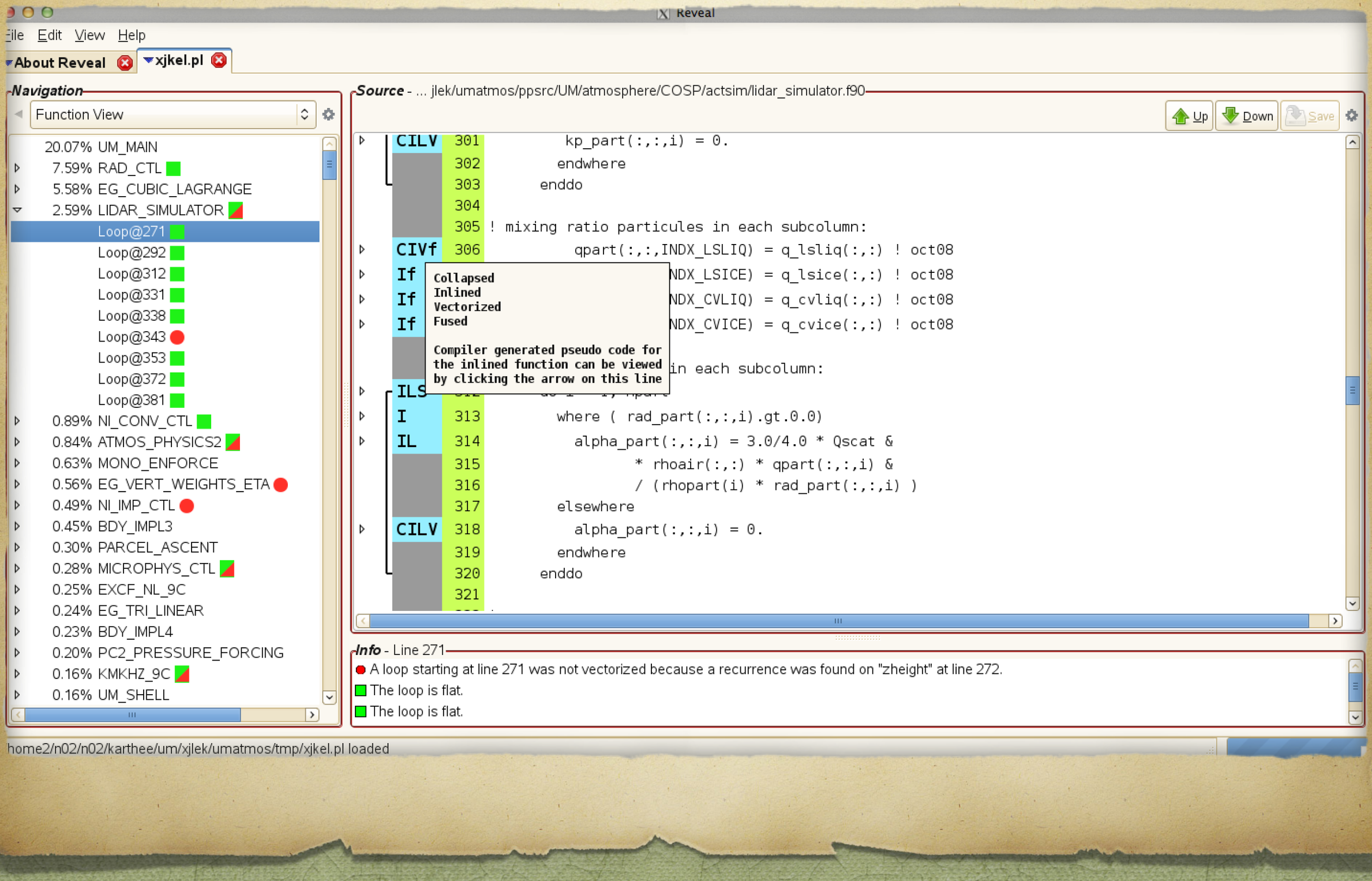
CRAY Reveal


Reveal
File Edit View Help
About Reveal
xjkel.pl
Navigation
Program View
atm_step_local_mod.f90
atm_step_phys_init.f90
atm_step_phys_reset.f90
atm_step_stash.f90
atm_step_swapbnds.f90
atm_step_timestep_init.f90
atmos_constants_mod.f90
atmos_max_sizes.f90
atmos_physics1.f90
atmos_physics2.f90
atmos_physics2_alloc.f90
augment_radiance.f90
augment_tiled_radiance.f90
backscatter_spectrum.f90
balance_lbc_values.f90
balance_lbc_values_4A.f90
EG_BALANCE_LBC_VALUES
Loop@279
Loop@280
Loop@286
Loop@287
Loop@294
Loop@295
Loop@302
Loop@303
Loop@309
Loop@310
Source - ... /umatmos/ppsrc/UM/atmosphere/lbc_input/balance_lbc_values_4A.f90
Up
Down
Save
447
448 ! NB. Broken into two separate loops to make vectorisable
449
ILS 450
451
ILS 452
ILS 453
454
455
456 !--------
457 ! Calculat
458 !--------
459
460
461
462
463
464
465
466
467
468
469
Reveal OpenMP Scoping
Scope Loops
Scoping Results
List of Loops to be Scoped
Scope? Line # File or Source Line
/home2/n02/n02/karthee/um/xjlek/umatmos/ppsrc/UM/at
279 DO k = tdims_s%k_start, tdims_s%k_end
280 DO i = 1, lenrim
286 DO k = tdims_s%k_start, tdims_s%k_end
287 DO i = 1, lenrim
294 DO k = tdims_s%k_start, tdims_s%k_end
295 DO i = 1, lenrim
302 DO k = tdims_s%k_start, tdims_s%k_end
303 DO i = 1, lenrim
309 DO k = tdims_s%k_start, tdims_s%k_end
310 DO i = 1, lenrim
324 DO j = pdims%j_start, pdims%j_end
Apply Filter
Time: 0.000
Trips: 0
Start Scoping
Cancel
Close
Info
home2/n02/n02/karthee/um/xjlek/umatmos/tmp/xjkel.pl loaded

CRAY Reveal


Reveal
File Edit View Help
About Reveal
xjkel.pl
Navigation
Program View
atm_step_local_mod.f90
atm_step_phys_init.f90
atm_step_phys_reset.f90
atm_step_stash.f90
atm_step_swapbnds.f90
atm_step_timestep_init.f90
atmos_constants_mod.f90
atmos_max_sizes.f90
atmos_physics1.f90
atmos_physics2.f90
atmos_physics2_alloc.f90
augment_radiance.f90
Source - ... /umatmos/ppsrc/UM/atmosphere/lbc_input/balance_lbc_values_4A.f90
Up
Down
Save
ILS 450 DO k = 1, model_levels-1
451 kp = k + 1
ILS 452 DO j = first_row, last_row
ILS 453 DO i = row_start_pt, row_end_pt
454 IF (ns_weights(i,j) /= 0.0) THEN
455
456 !---
457 ! Calculate new balanced values of Exner
458 !---
459
460 iloc = lbc_address(i,j)
work1(i,j,kp) = 0.5*(u_lbc(lbc_addres
+ u_lbc(lbc_address
*f2_comp_eh(i,j)
- 0.5*(v_lbc(lbc_addres
+ v_lbc(lbc_address
*f1_comp_eh(i,j)
denom1 = cp * theta_wet_lbc(iloc,k) /
(r_rho_levels(i,j,kp) - r_rh
vectorized because a recurrence was found on "exner_lb
The loop is flat.
OpenMP Directive
! Directive inserted by Cray Reveal. May be incomplete.
!$OMP parallel do default(none) &
!$OMP& unresolved (K,KP,ROW_END_PT,LAST_ROW) &
!$OMP& private (i,j,iloc,v_av,u_av,denom1,metric_term,dpdz,temp) &
!$OMP& shared (lenrim,exner_lbc,model_levels,rho_lbc,theta_lbc,w_lbc, &
!$OMP& lenrimu,u_lbc,lenrimv,v_lbc,r_rho_levels, &
!$OMP& r_theta_levels,row_length,halo_i,halo_j,rimwidth, &
!$OMP& f1_comp_eh,f2_comp_eh,g_theta_eh,intw_rho2w, &
!$OMP& intw_w2rho,lbc_address,lbc_address_im1, &
!$OMP& lbc_address_jm1,lbc_address_u,lbc_address_v, &
!$OMP& l_cartesian,ns_weights,theta_wet_lbc,work1)
Copy Directive
Close
Reveal OpenMP Scoping
Scope Loops
Scoping Results
balance_lbc_values_4A.f90: Loop@452
Name Type Scope Info
K Scalar Unresolved FAIL: No scoping information
KP Scalar Unresolved FAIL: No scoping information
LAST_ROW Scalar Unresolved FAIL: No scoping information
ROW_END_PT Scalar Unresolved FAIL: No scoping information
denom1 Scalar Private
dpdz Scalar Private
i Scalar Private
iloc Scalar Private
j Scalar Private
metric_term Scalar Private
First/Last Private
Enable FirstPrivate
Enable LastPrivate
Reduction
None
Find Name:
Insert Directive
Show Directive
home2/n02/n02/karthee/um/xjlek/umatmos/tmp/xjkel.pl loaded

CRAY Reveal


Reveal
File Edit View Help
About Reveal
xjkel.pl
Navigation
Function View
20.07% UM_MAIN
7.59% RAD_CTL
5.58% EG_CUBIC_LAGRANGE
2.59% LIDAR_SIMULATOR
Loop@271
Loop@292
Loop@312
Loop@331
Loop@338
Loop@343
Loop@353
Loop@372
Loop@381
0.89% NI_CONV_CTL
0.84% ATMOS_PHYSICS2
0.63% MONO_ENFORCE
0.56% EG_VERT_WEIGHTS_ETA
0.49% NI_IMP_CTL
0.45% BDY_IMPL3
0.30% PARCEL_ASCENT
0.28% MICROPHYS_CTL
0.25% EXCF_NL_9C
0.24% EG_TRI_LINEAR
0.23% BDY_IMPL4
0.20% PC2_PRESSURE_FORCING
0.16% KMKHZ_9C
0.16% UM_SHELL
Source - ... jlek/umatmos/ppsrc/UM/atmosphere/COSP/actsim/lidar_simulator.f90
Up
Down
Save
CILV 301 kp_part(:,:,i) = 0.
302 endwhere
303 enddo
304
305 ! mixing ratio particules in each subcolumn:
CIVf 306 qpart(:,:,INDX_LSLIQ) = q_lsliq(:,:) ! oct08
If NDX_LSICE) = q_lsice(:,:) ! oct08
If NDX_CVLIQ) = q_cvliq(:,:) ! oct08
If NDX_CVICE) = q_cvice(:,:) ! oct08
in each subcolumn:
ILS
I 313 where (rad_part(:,:,i).gt.0.0)
IL 314 alpha_part(:,:,i) = 3.0/4.0 * Qscat &
315 * rhoair(:,:) * qpart(:,:,i) &
316 / (rhopart(i) * rad_part(:,:,i))
317 elsewhere
CILV 318 alpha_part(:,:,i) = 0.
319 endwhere
320 enddo
321
Collapsed
Inlined
Vectorized
Fused
Compiler generated pseudo code for the inlined function can be viewed by clicking the arrow on this line
Info - Line 271
A loop starting at line 271 was not vectorized because a recurrence was found on "zheight" at line 272.
The loop is flat.
The loop is flat.
home2/n02/n02/karthee/um/xjlek/umatmos/tmp/xjkel.pl loaded

CRAY Reveal

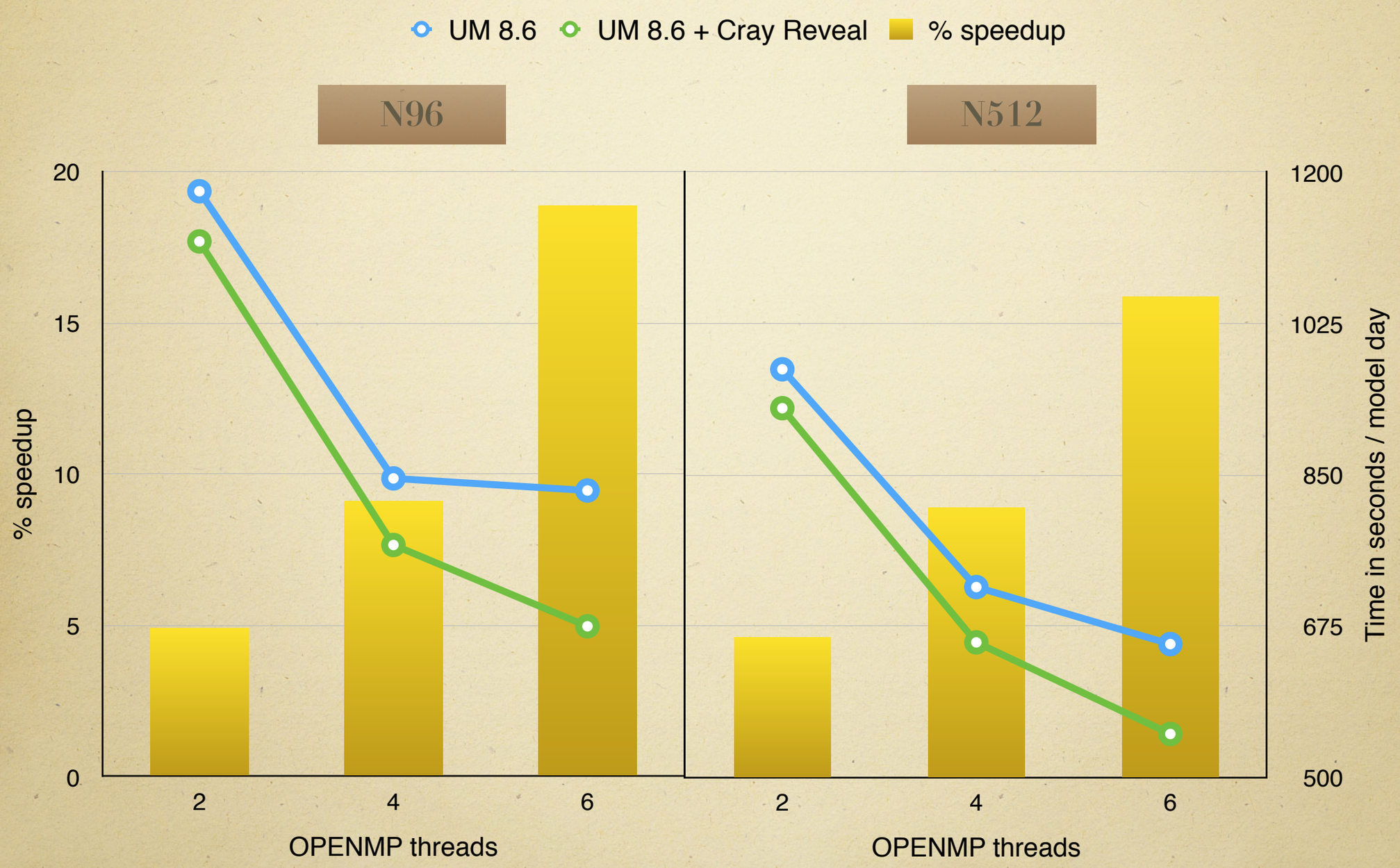
- 2389 serial loops are parallelised using OPENMP directives as suggested by CRAY Reveal
- Fortran array notation expressions , for all and where statements are parallelised using

```
!$OMP PARALLEL WORKSHARE

!$OMP END PARALLEL WORKSHARE
```

- Bit reproducibility is tested for correctness of OPENMP directives.
- default(none) option is used for all the newly added directives.

Performance speedup



Conclusions

- ARCHER (Cray XC30) has good MPI bandwidth and supports hardware threads
- CrayPAT tools used to analyse the performance
- MPI tuning result in 5 to 12% speedup
- UM has poor thread load balance.
- CRAY Reveal tool used to parallelise serial loops
- New OPENMP directives improve performance by 5% up to 20% when using 2 to 6 threads.

THANK YOU

- ARCHER (Cray XC30) has good MPI bandwidth and supports hardware threads
- CrayPAT tools used to analyse the performance
- MPI tuning result in 5 to 12% speedup
- UM has poor thread load balance.
- CRAY Reveal tool used to parallelise serial loops
- New OPENMP directives improve performance by 5% up to 20% when using 2 to 6 threads.