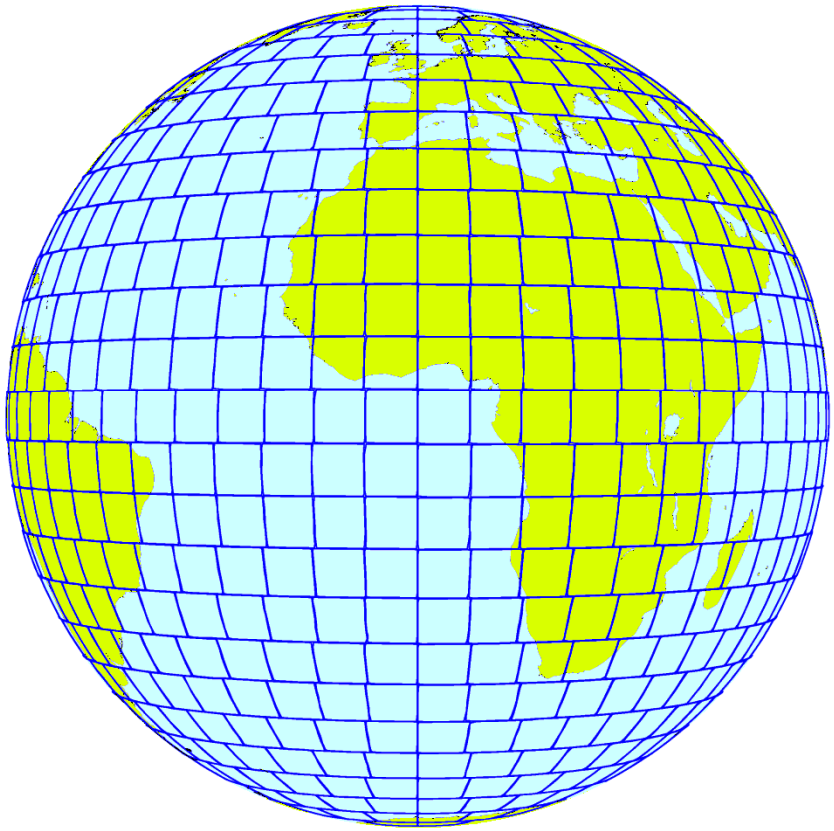


IFS: RAPS11 and model scaling

George Mozdzynski

ECMWF

Outline

- **IFS RAPS11 benchmark**
- **Scaling IFS model on Power6**
  - **Speedup and Efficiency**
  - **Scaling problem areas**
  - **Weak scaling**
- **Scaling to 100K – 1M threads**
  - **Rewrite or optimise our applications**
  - **Parallel programming languages**
  - **What do we need of future HPC systems**

IFS RAPS11 benchmark

- **Released January 2010, CY36R2**
- **IFS model only (no 4D-Var)**
  - **Sources cut down to called routines**
- **What's new in RAPS11**
  - **grib_api (supporting grib1 and grib2 standards)**
  - **T1279L149 (~16 km)**
  - **T2047L149 (~10 km)**
  - **Internet distribution (& USB disk)**
    - **T2047L149 data is 13 GB**
- **Updates (at raps@ftp.ecmwf.int)**
  - **4 in total, last update added 8th Oct**

IFS RAPS11 benchmark

- **Model resolutions T159, T399, T799, T1279 and T2049**
- **Full outputs from IBM Power6 (all resolutions)**
- **No output of model fields (i.e. not an I/O benchmark)**
- **Reference job scripts**
  - **24 time steps**
  - **test of correctness**

```
Results of ERROR calculation

The error calculated from the results shows
that the calculations are correct

The maximum error is =          0.16518 %
```

- **Long run job scripts, for performance runs**
  - **use same executable as for reference runs**
- **Norms (SP,GP) should be bit-reproducible when changing**
  - **number of MPI tasks**
  - **number of OpenMP threads**

IFS RAPS11 benchmark **DR_HOOK_OPT=prof** output provided for all model cases (T2047 extract)

```
Name of the executable : /fws2/lb/work/rd/mpm/RAPS11/T2047/../bin/ifsMASTER
Number of MPI-tasks : 512
Number of OpenMP-threads : 8
Wall-times over all MPI-tasks (secs) : Min=1020.070, Max=1025.730, Avg=1022.972, StDev=1.261
Routines whose total time (i.e. sum) > 1.000 secs will be included in the listing
  Avg-%   Avg.time   Min.time   Max.time   St.dev  Imbal-%   # of calls : Name of the routine
  7.91%     80.951     71.799     86.287    1.554   16.79%      6545408 : MXMAOP
  4.15%     42.488     39.259     47.132    1.530   16.70%     87748608 : CLOUDSC
  3.65%     37.345     31.398     43.424    1.999   27.69%  44742137979 : CUADJTQ
  3.62%     37.059     36.473     38.821    0.240    6.05%    350994432 : LAITRI
  3.59%     36.672     35.747     38.017    0.342    5.97%    482818560 : VERINT
  3.12%     31.937     27.864     36.514    1.297   23.69%       100352 : >MPL-TRMTOL_COMMS(807)
  3.03%     30.991     15.474     70.559   10.875   78.07%        98816 : >MPL-TRLTOM_COMMS(806)
  2.93%     29.964      3.977     71.212   12.580   94.42%        98816 : >MPL-TRGTOL_COMMS(803)
  2.45%     25.066     12.663     37.822    5.504   66.52%         1611 : >MPL-IOSTREAMREAD_RECORD(650)
  2.43%     24.853      8.287     33.075    3.370   74.94%       100352 : >MPL-TRLTOG_COMMS(805)
  2.21%     22.597     21.537     23.729    0.336    9.24%    175497216 : LASCAW
  2.03%     20.776     20.174     21.547    0.242    6.37%     87748608 : VDFMAIN
  1.85%     18.952     18.393     20.062    0.245    8.32%    789737472 : LAITLI
  1.70%     17.341     16.440     18.936    0.364   13.18%     87748608 : VDFEXCU
  1.55%     15.812     13.110     22.851    1.057   42.63%        98816 : >BAR-BARRIERINSIGCHECK(718)
  1.49%     15.241      6.446     25.483    4.342   74.70%       802816 : >OMP-FTINV_CTL(1639)
  1.40%     14.324      9.447     23.023    1.886   58.97%        99840 : >MPL-SLCOMM1_COMMS(509)
  1.39%     14.183     13.788     14.387    0.076    4.16%    175497216 : LARCHE
  1.37%     14.056      8.137     25.275    2.968   67.81%     43874304 : CUBASEN
  1.18%     12.059      1.922     25.705    5.039   92.52%     43874304 : CLOUDVAR
  1.04%     10.594      6.716     16.324    1.017   58.86%       796672 : >OMP-LTINV_CTL-INVERSE(1647)
  1.03%     10.583     10.194     11.688    0.258   12.78%       786432 : >OMP-WAMODEL2(1431)
  0.97%      9.930      0.000     18.834    4.970  100.00%         1024 : >MPL-BROADCASTIOSTREAMGR(632)
  0.94%      9.637      9.043     10.247    0.191   11.75%       790528 : >OMP-CPG1(1025)
  0.93%      9.484      9.059      9.978    0.152    9.21%      1702400 : SRTM_SPCVRT_MCICA
  0.90%      9.210      7.612     11.005    0.545   30.83%       790528 : >OMP-LTDIR_CTL-DIRECT(1645)
  0.90%      9.193      8.827      9.538    0.145    7.45%     43874304 : CPDYDDH
```


ECMWF

Outline

- **IFS RAPS11 benchmark**
- → **Scaling IFS model on Power6**
  - **Speedup and Efficiency**
  - **Scaling problem areas**
  - **Weak scaling**
- **Scaling to 100K – 1M threads**
  - **Rewrite or optimise our applications**
  - **Parallel programming languages**
  - **What do we need of future HPC systems**

Power5: SpeedUp and Efficiency
T799L91 model, 2 day forecast (CY33R1)

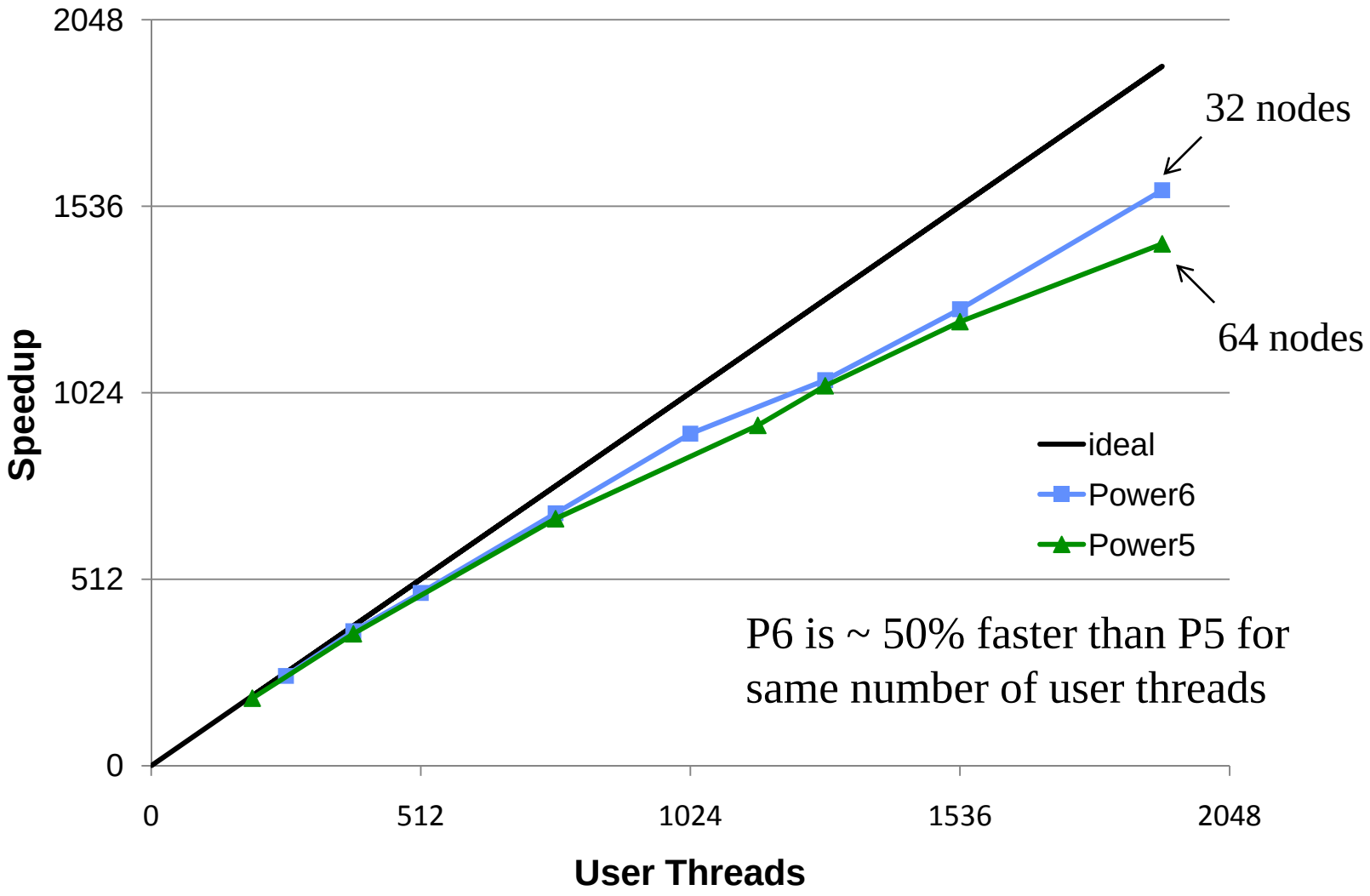
November 2008

parallel	serial
649849	115

User Threads = MPI tasks x OpenMP threads

User Threads	Actual Wall Time	Calculated Wall Time	Calculated SpeedUp	Calculated Efficiency %
192	3505.3	3500	185	96.6
384	1794.6	1808	362	94.3
768	958.5	962	678	88.3
1152	695.7	680	934	81.1
1280	623.2	623	1043	81.5
1536	533.2	539	1219	79.4
1920	453.7	454	1433	74.6
1		649965	**10 day forecast ~ 45 min**	

T799L91 model



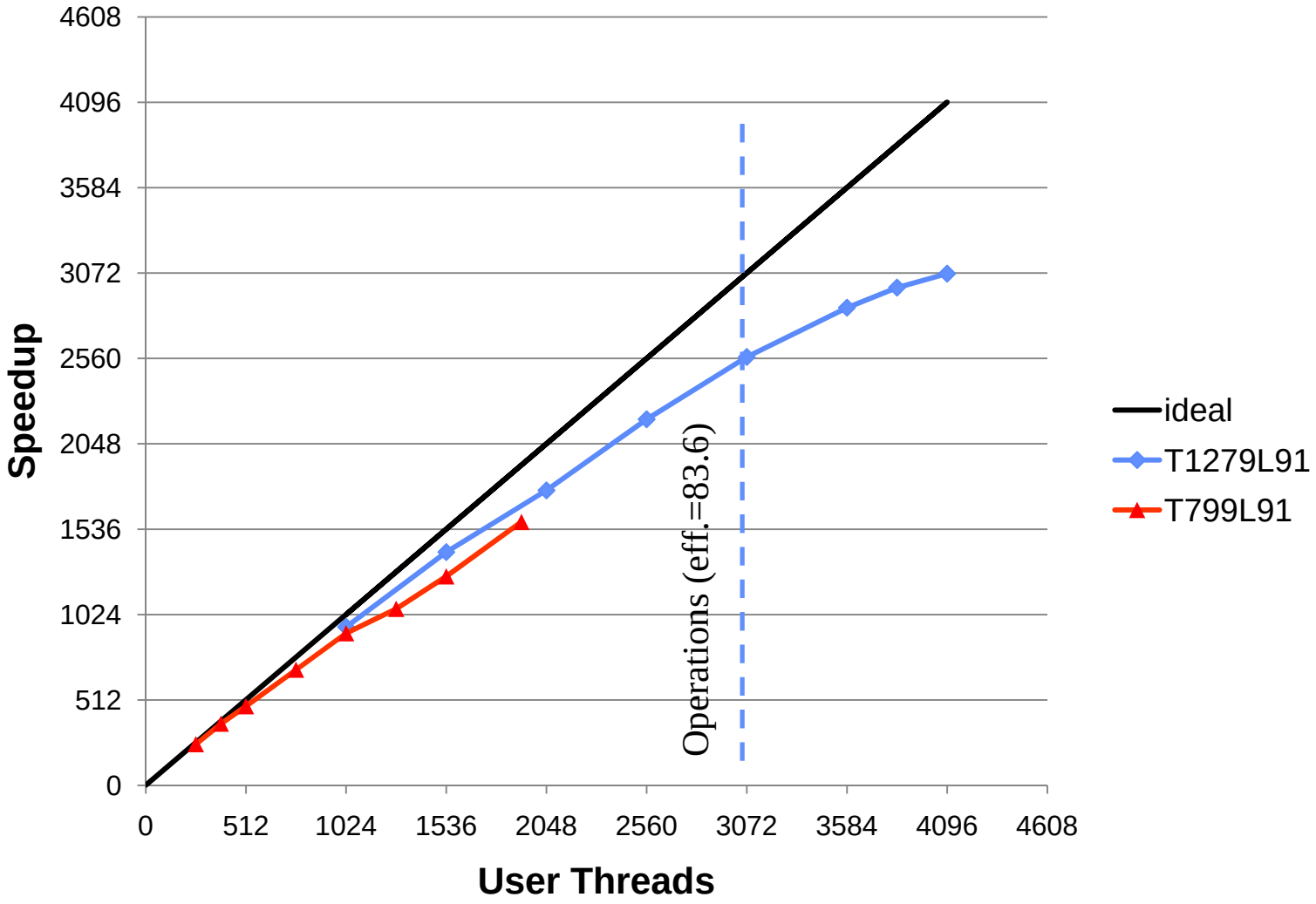
Power6: SpeedUp and Efficiency
T1279L91 model, 2 day forecast (CY36R4)

parallel	serial
1591806	114

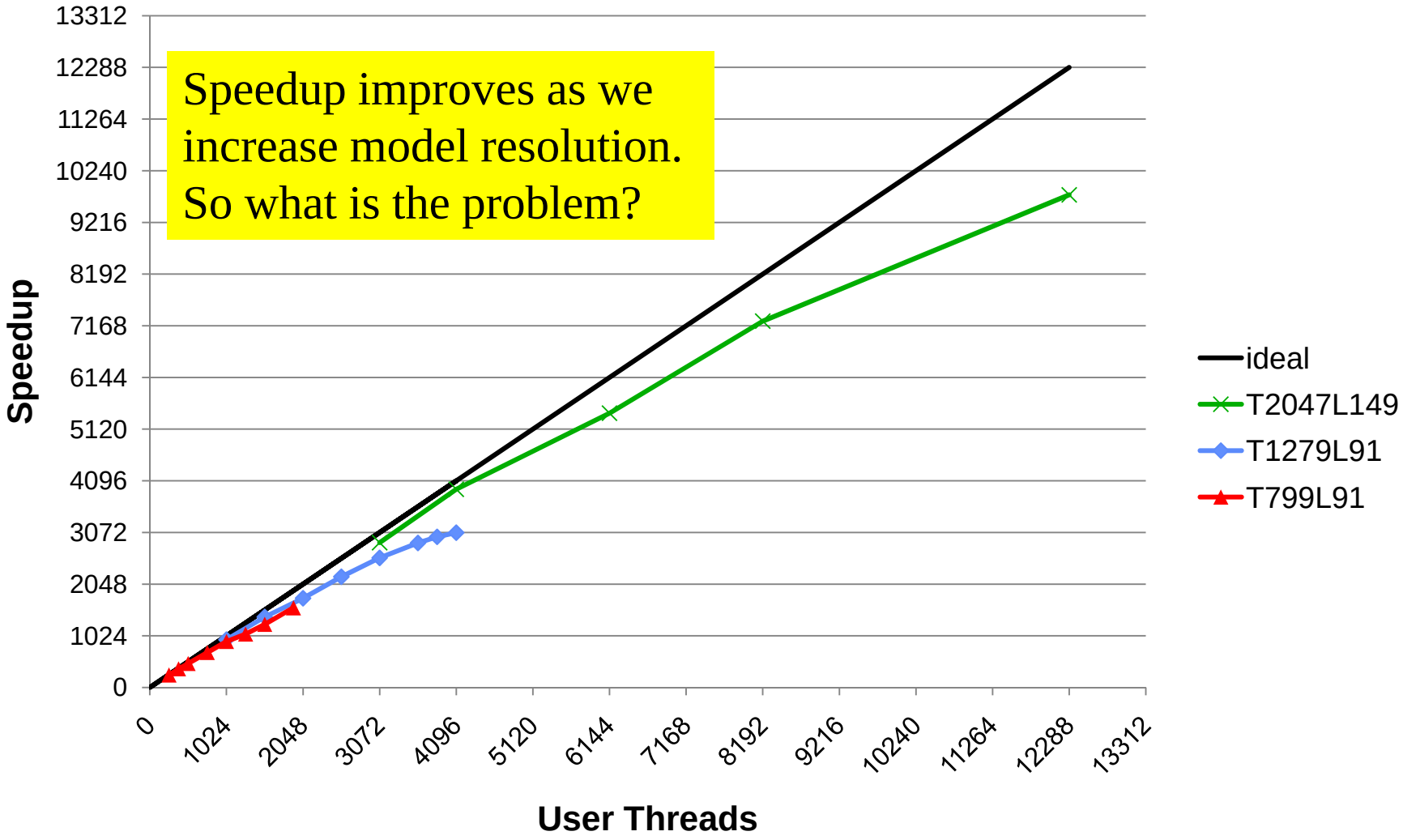
User Threads	Actual Wall Time	Calculated Wall Time	Calculated SpeedUp	Calculated Efficiency %
1024	1675.7	1668	950	92.8
1536	1138.0	1150	1399	91.1
2048	899.9	891	1769	86.4
2560	725.1	736	2195	85.8
3072	619.7	632	2569	83.6
3584	555.8	558	2864	79.9
3840	533.3	528	2985	77.7
4096	518.8	502	3068	74.9
1		1587754	**10 day forecast ~ 45 min**	

ECMWF

IFS model speedup on Power6 (8 threads per task)



IFS speedup on Power6 (+ T2047L149 RAPS11)



HPC systems: near future

- **Approaching limits in single core performance**
  - **Clock frequency, Cooling, Cost of electricity**
- **Trend**
  - **Increase in # cores (cores per socket)**
  - **Little change in single core performance**
- **How will this affect NWP models such as IFS?**
  - **Requirement: 10 day forecast in < 45 mins**
  - **Increasing model resolution**
    - **Good for scaling: more parallelism to exploit**
    - **Bad for scaling: more time-steps to execute**

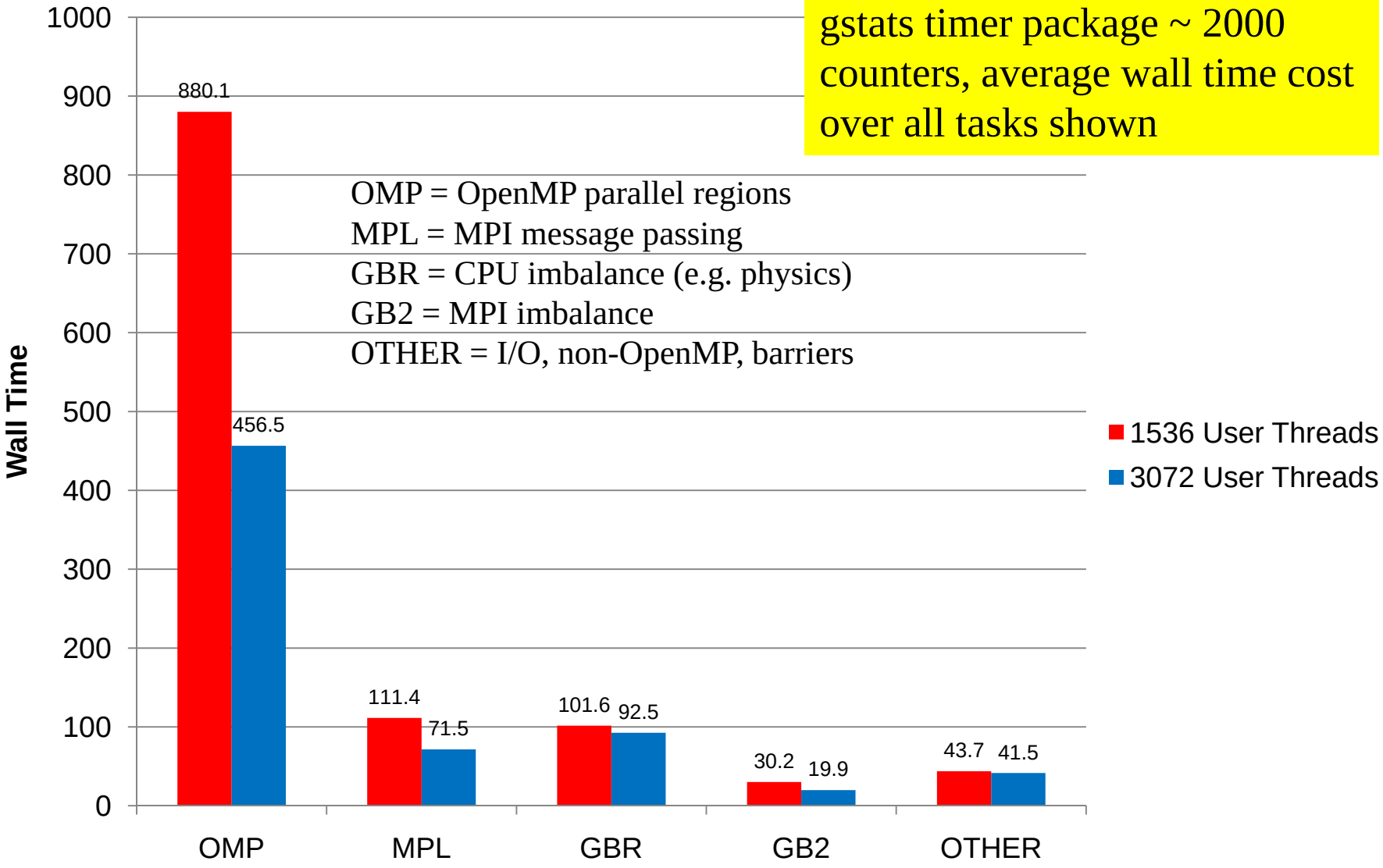
Weak scaling: IFS model, 10 day forecasts in ~ 45 minutes on Power6

Model Resolution	Time step (sec)	User Threads	Grid points per thread	Efficiency %
T799L91	720	1024	824	88
T1279L91	600	3840	557	78
T2047L149	450	18000**	303**	75**

** - extrapolation based on runs up to 12288 User Threads (192 P6 nodes)

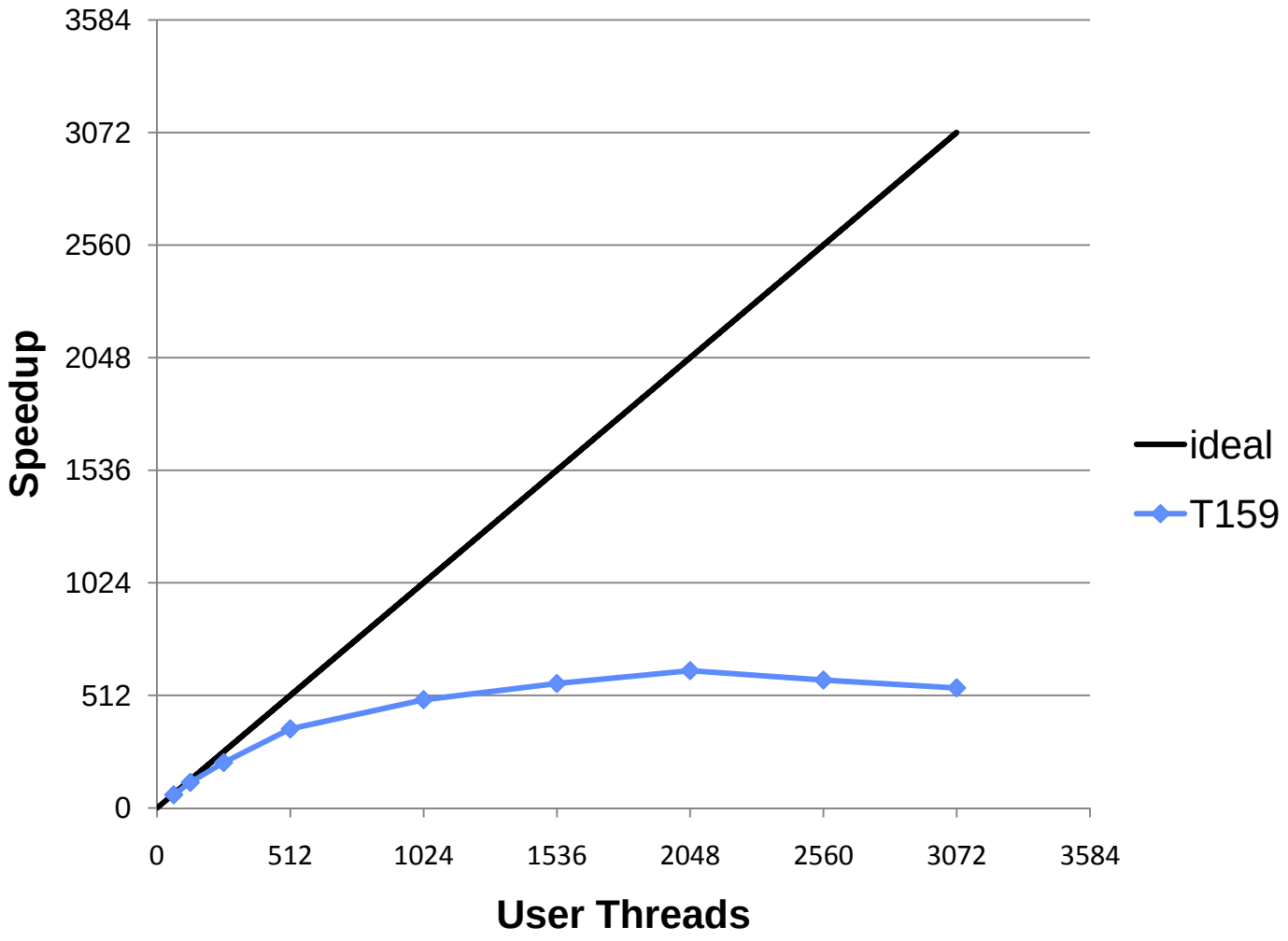
T1279 model – gstats counter summary

gstats timer package ~ 2000 counters, average wall time cost over all tasks shown

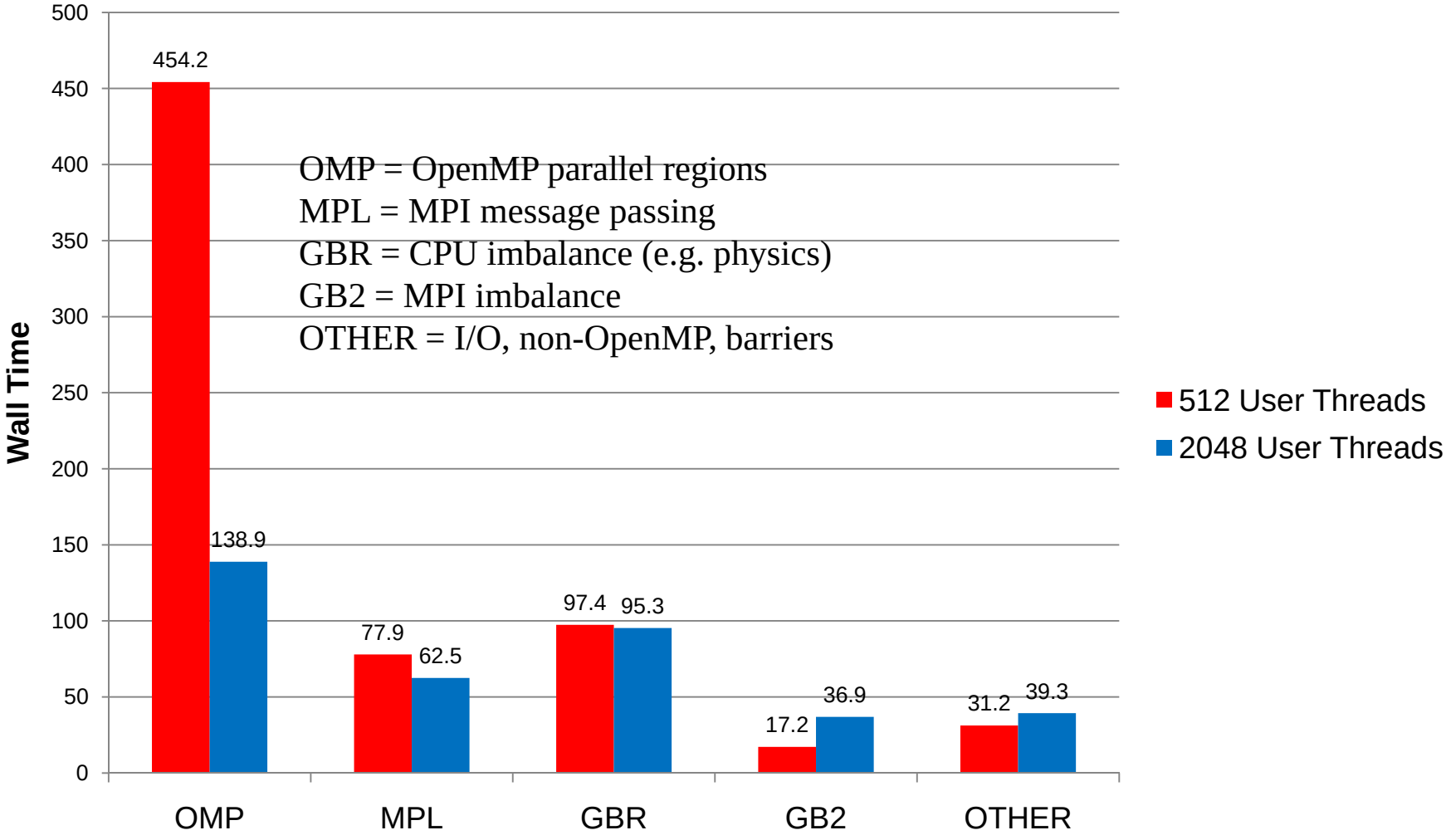


ECMWF

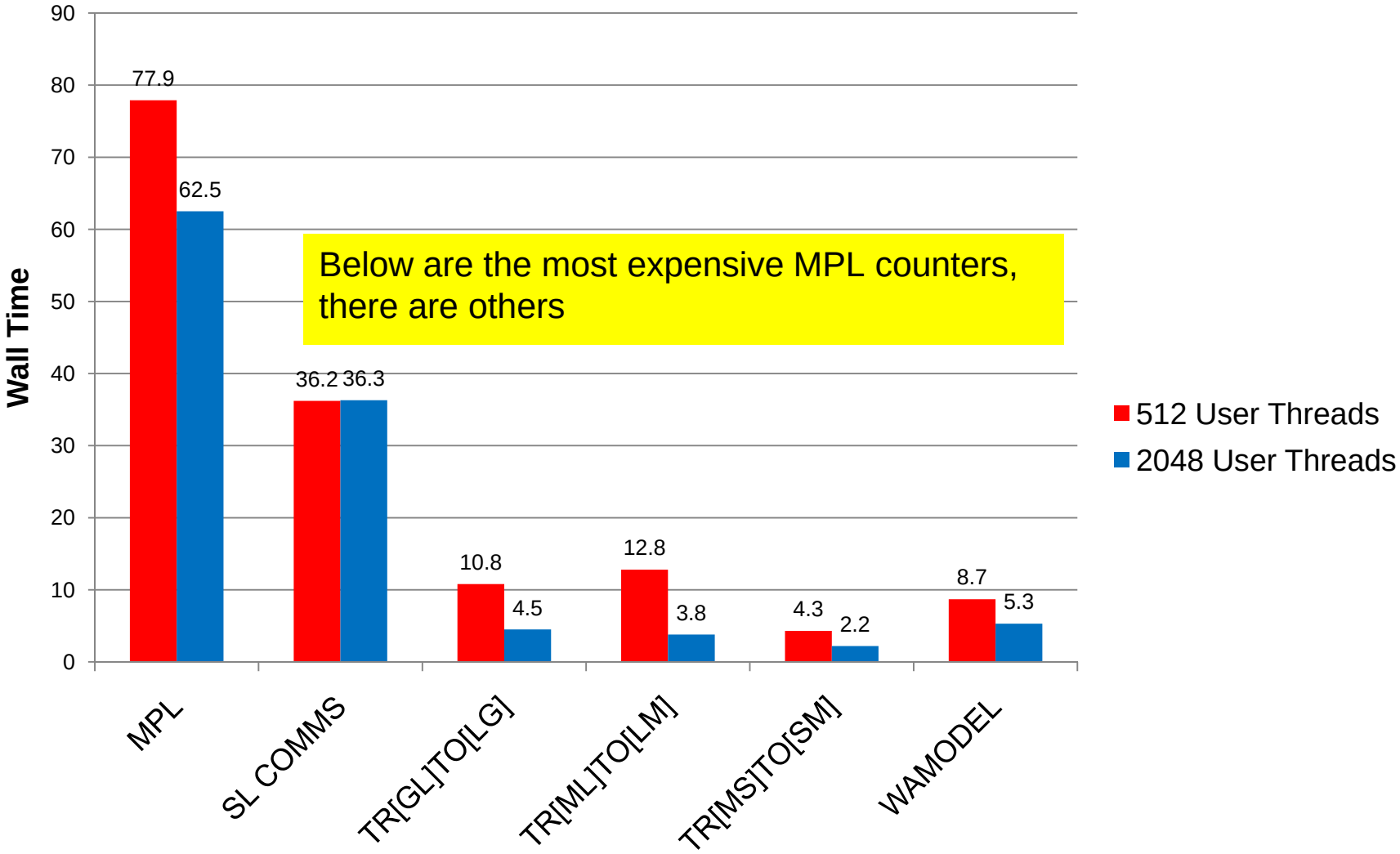
T159 model scaling: small model with ‘large’ number of user threads (4 threads per task)



T159 model – gstats counter summary

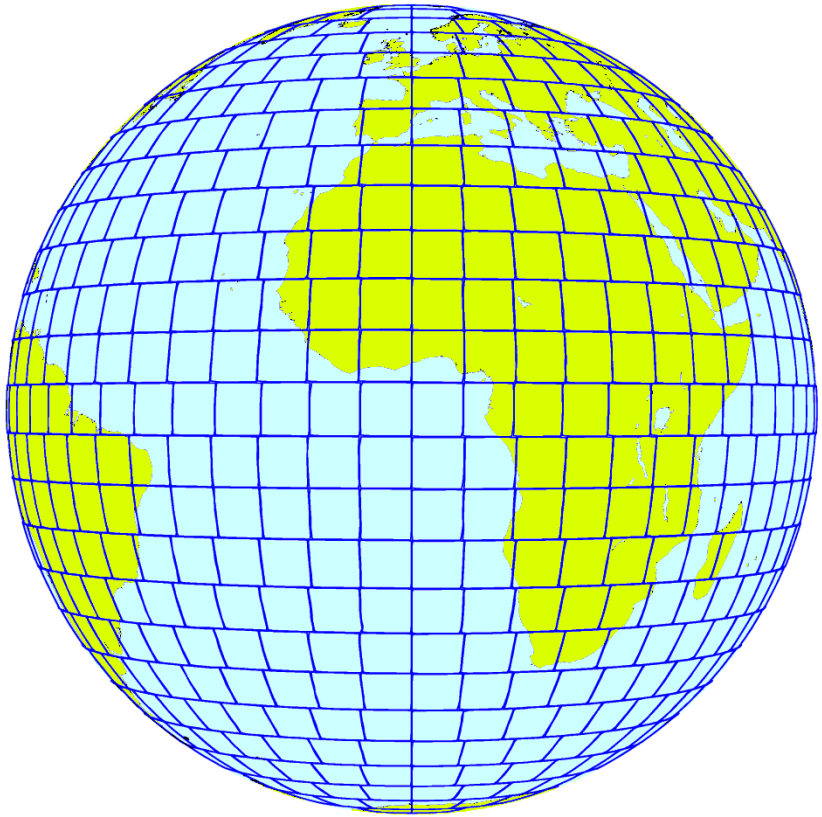


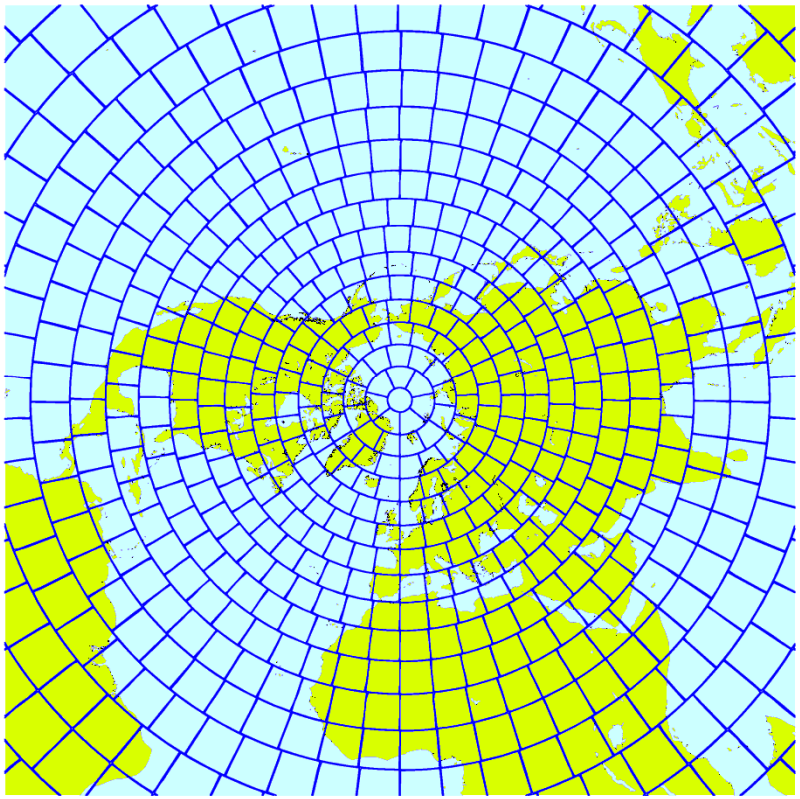
T159 model – MPL gstats counters



IFS grid point space: partitioning for 1024 MPI tasks

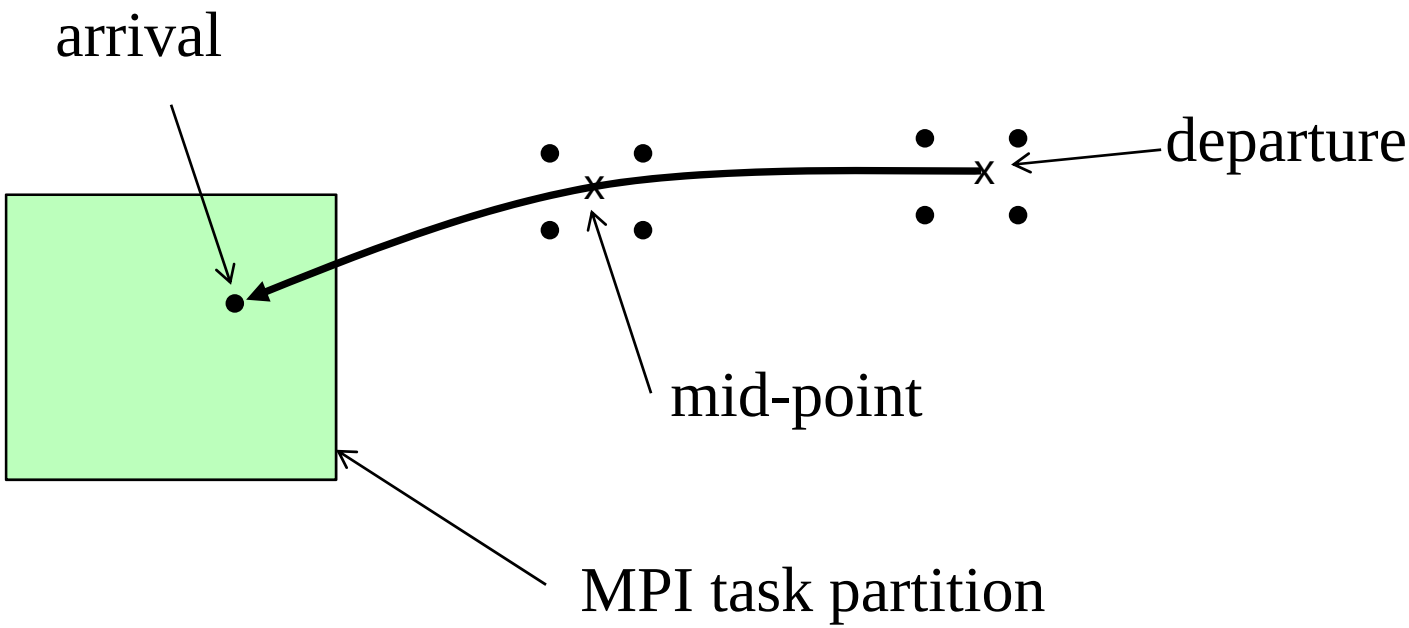
Each MPI task has an equal number of grid points



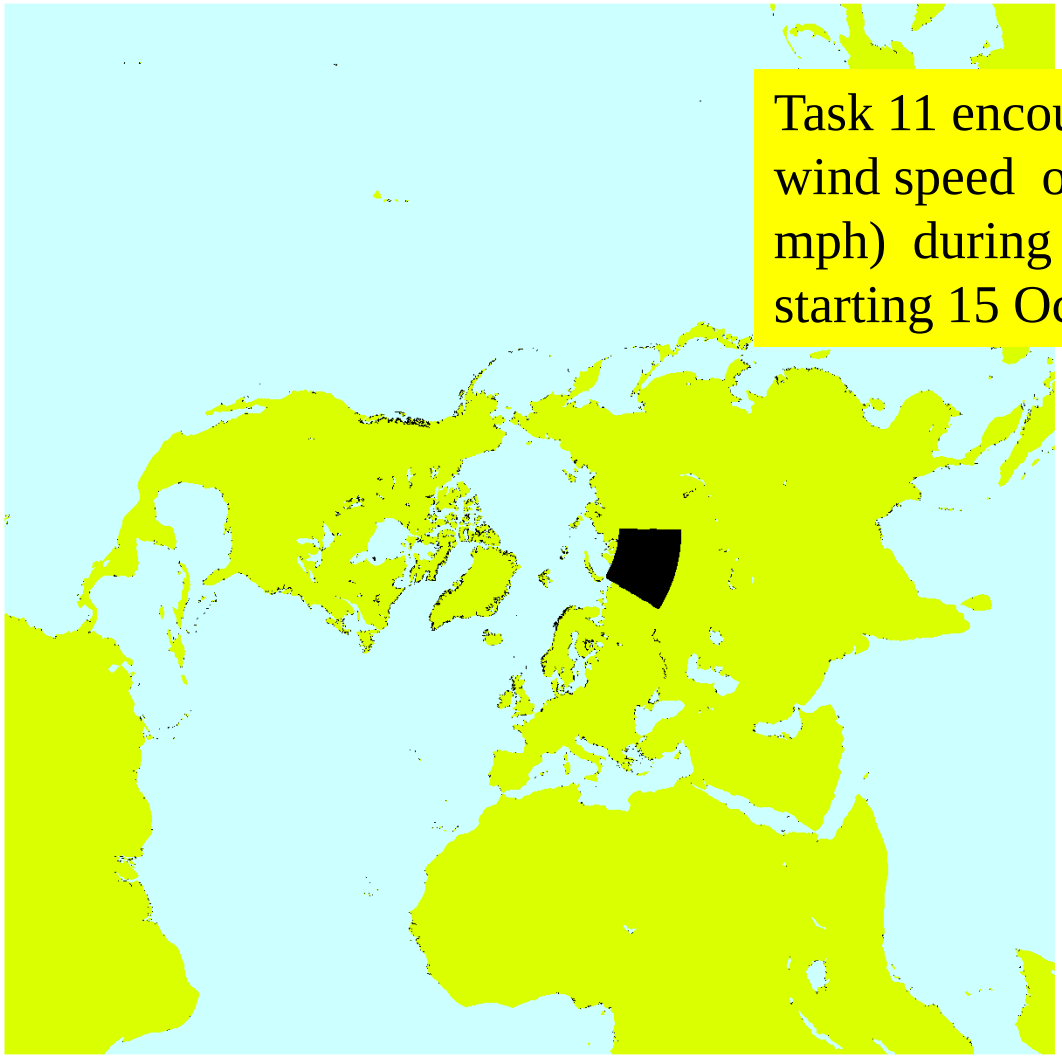


Semi-Lagrangian Transport

- **Computation of a trajectory from each grid-point backwards in time, and**
- **Interpolation of various quantities at the departure and at the mid-point of the trajectory**



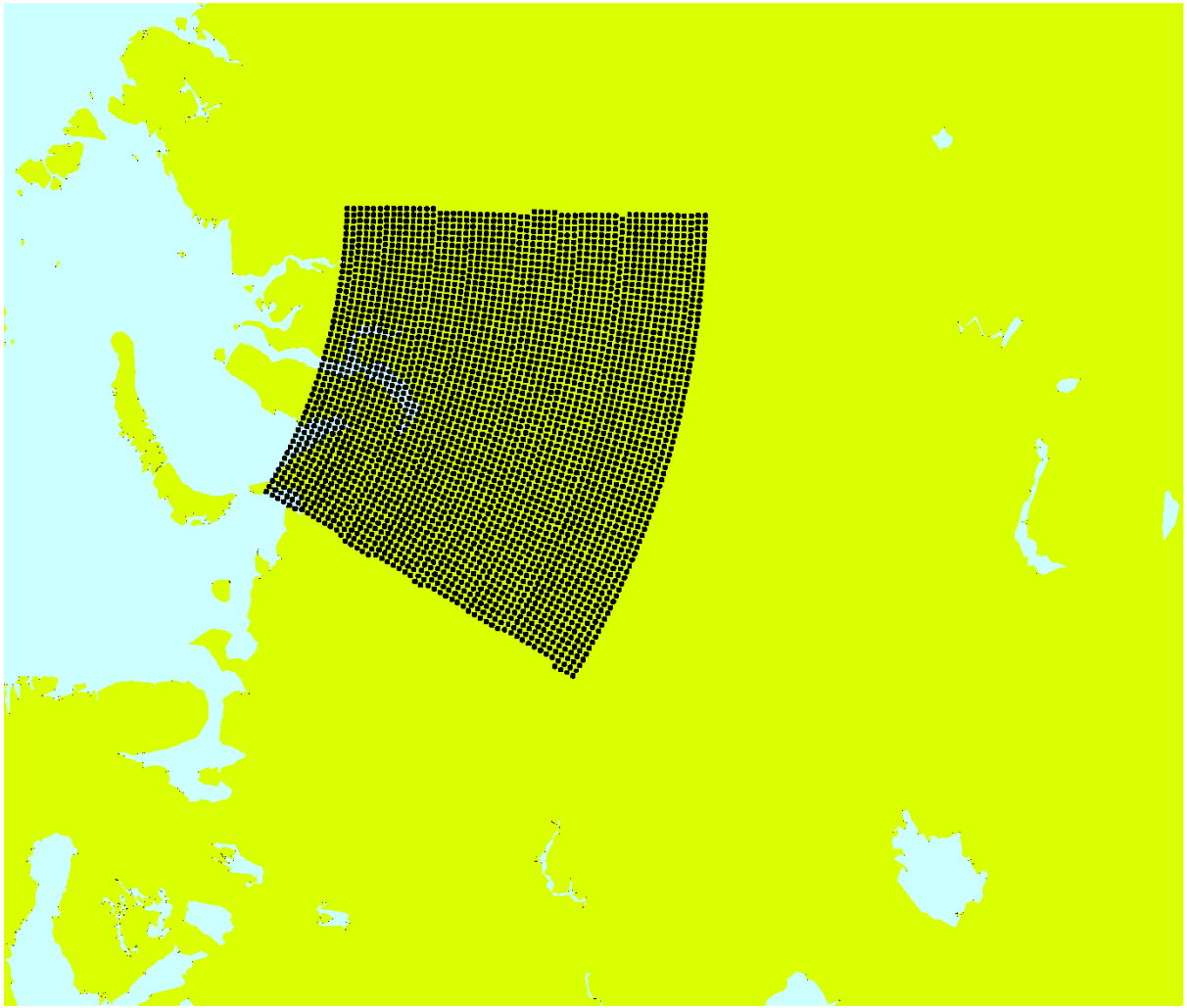
Semi-Lagrangian Transport: T799 model, 256 tasks



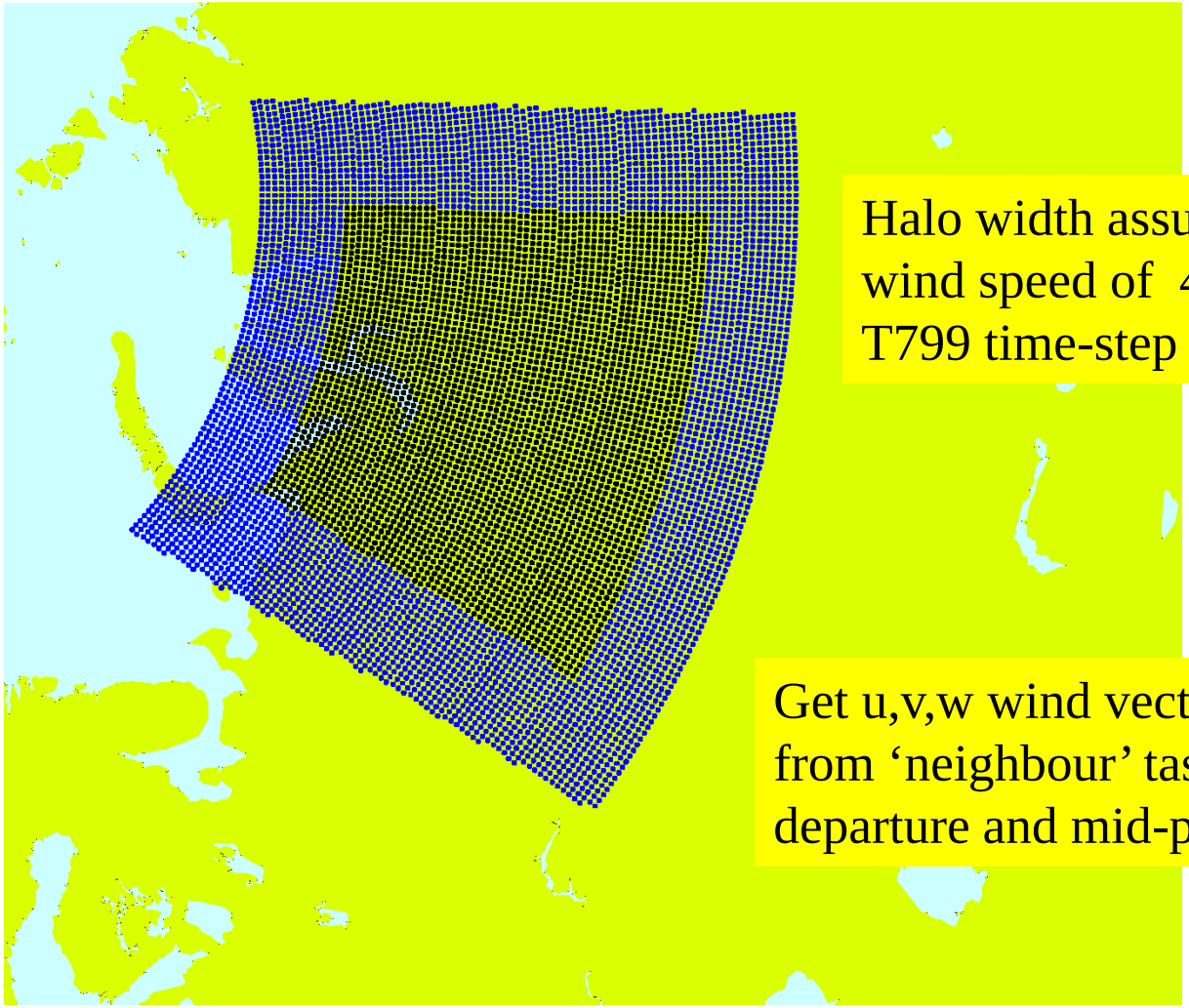
Task 11 encountered the highest wind speed of 120 m/s (268 mph) during a 10 day forecast starting 15 Oct 2004

T799 model task 11 of 256

black: grid points owned by task 11



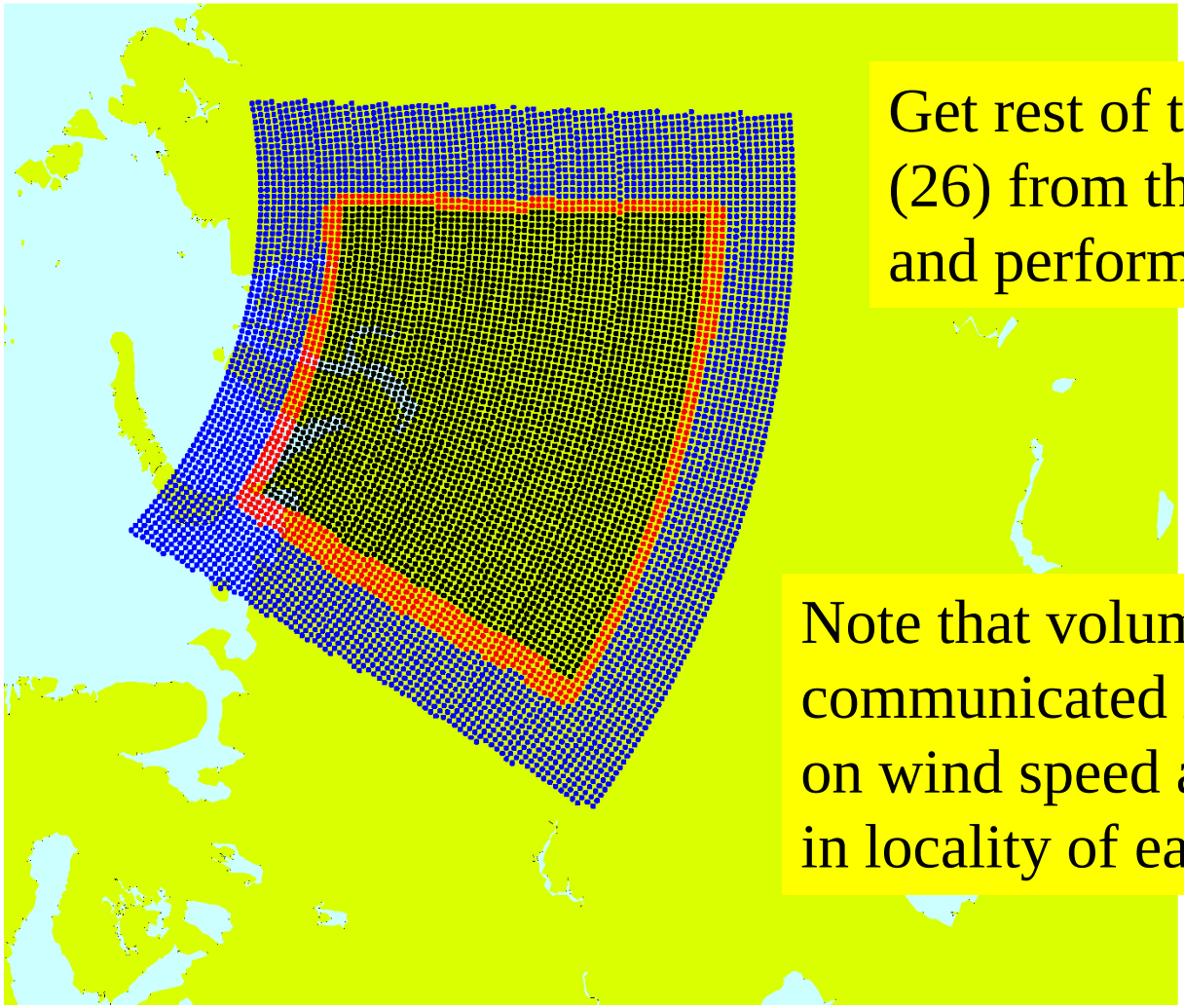
blue: halo area



Halo width assumes a maximum wind speed of 400 m/s x 720 s T799 time-step (288 km)

Get u,v,w wind vector variables (3) from ‘neighbour’ tasks to determine departure and mid-point of trajectory

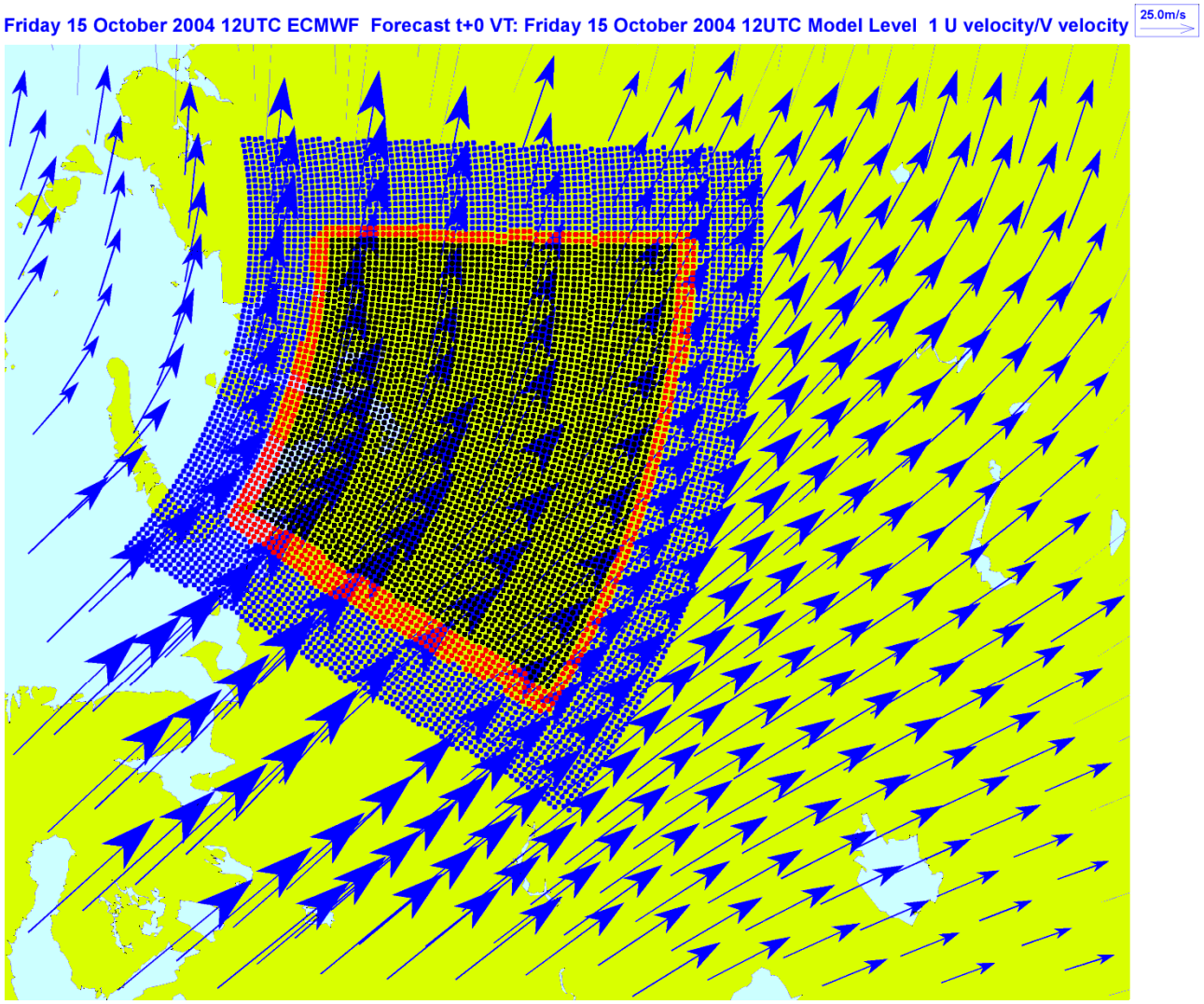
red: halo points actually used



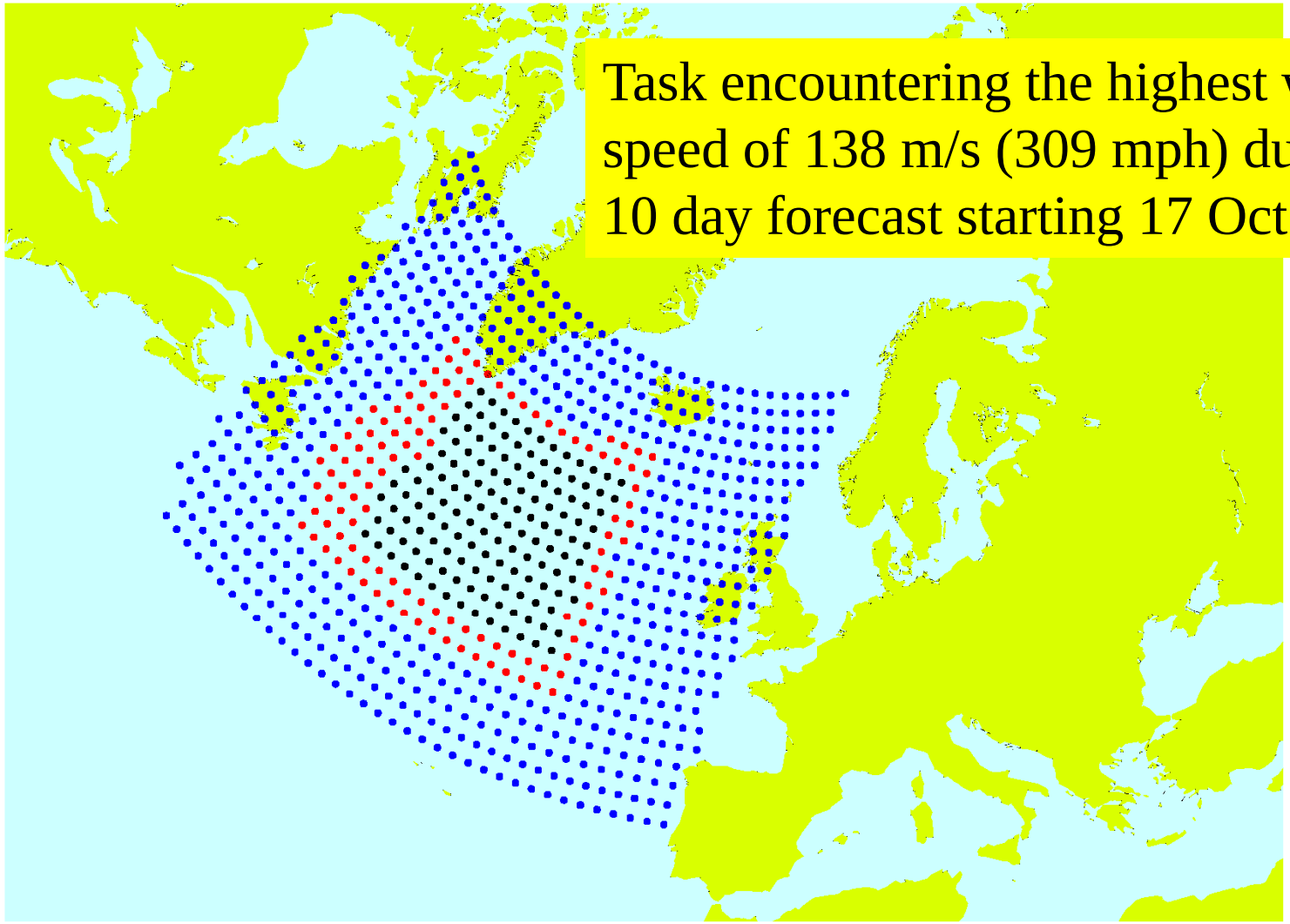
Get rest of the variables (26) from the red halo area and perform interpolations

Note that volume of halo data communicated is dependent on wind speed and direction in locality of each task

wind plot

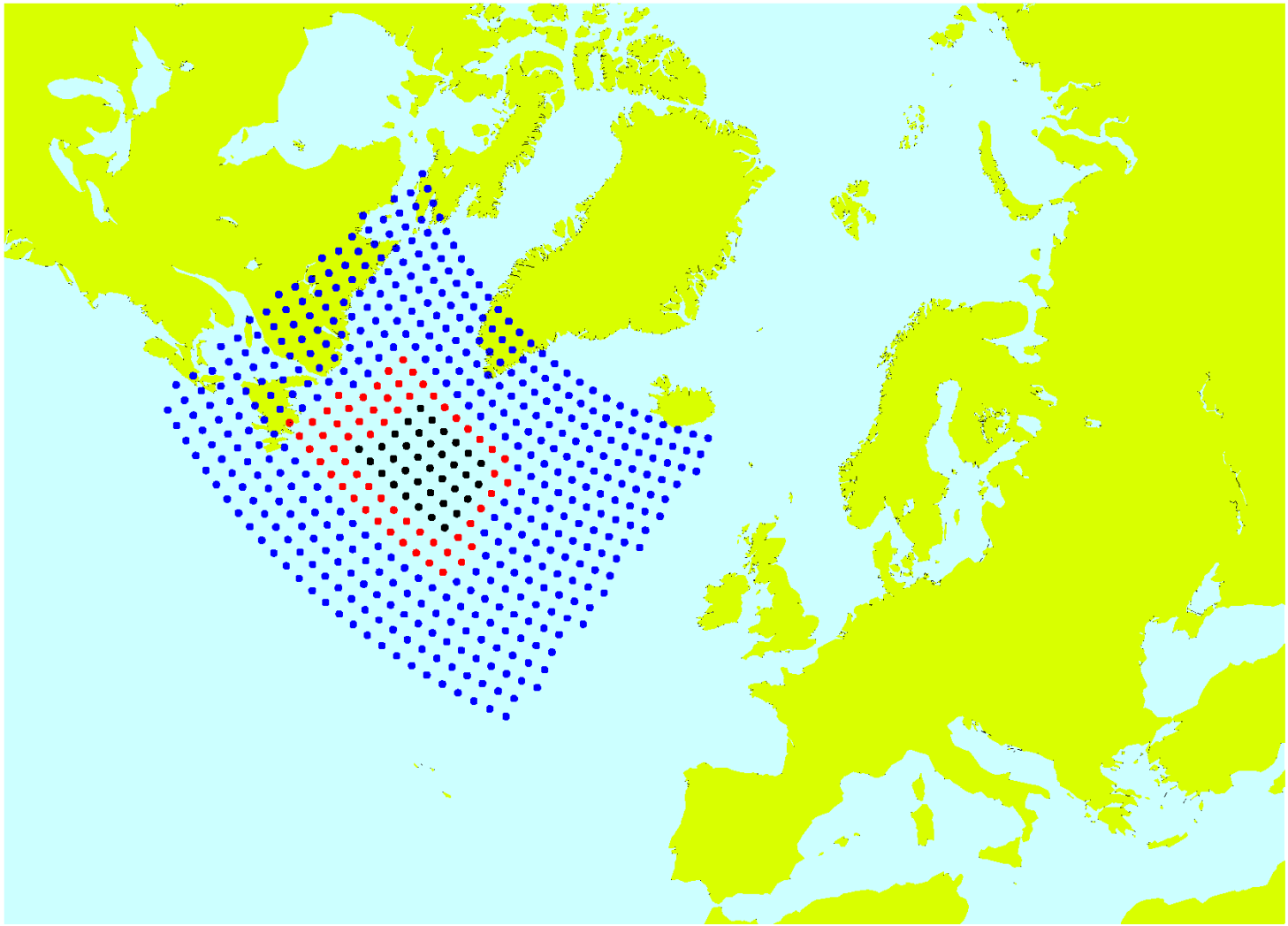

Friday 15 October 2004 12UTC ECMWF Forecast t+0 VT: Friday 15 October 2004 12UTC Model Level 1 U velocity/V velocity
25.0m/s

T159 model task 37 of 256 tasks

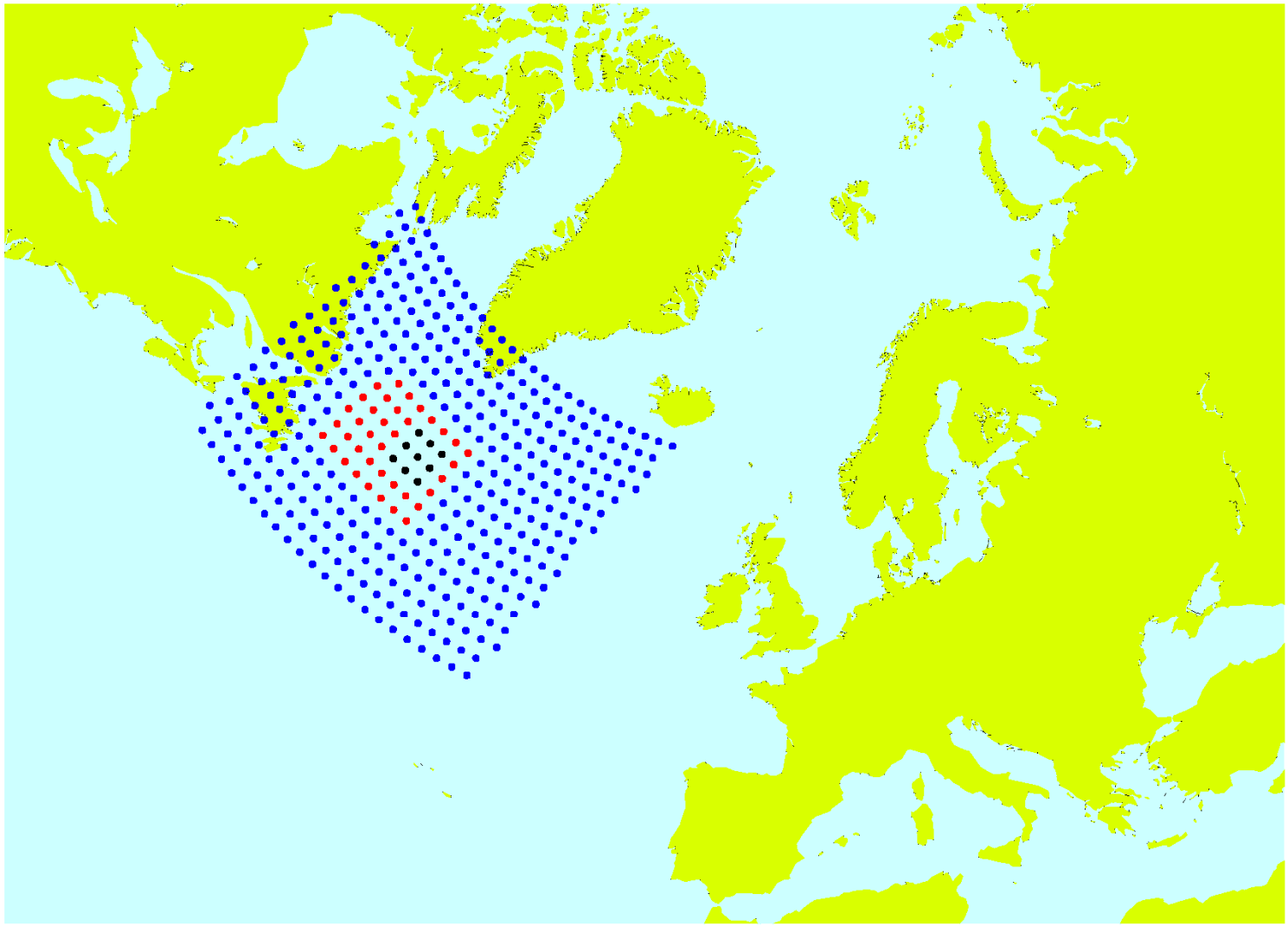


Task encountering the highest wind speed of 138 m/s (309 mph) during a 10 day forecast starting 17 Oct 2010

T159 model task 128 of 1024 tasks



T159 model task 462 of 4096 tasks



IFS Semi-Lagrangian Comms

- **SL comms scaling limited by**
  - **constant width halo for u,v,w (400 m/s x time step)**
  - **Halo volume communicated, which is a function of wind speed and direction in locality of each task**
- **‘Halo-lite’ approach tested**
  - **Only get (using MPI) grid columns from neighbour tasks that your task needs, i.e. only the red points**
  - **Requires more MPI communication steps (e.g. mid-point, departure point)**
  - **No faster than current approach due to overheads of above**
  - **Potential for optimisation using coarrays (F2008)**

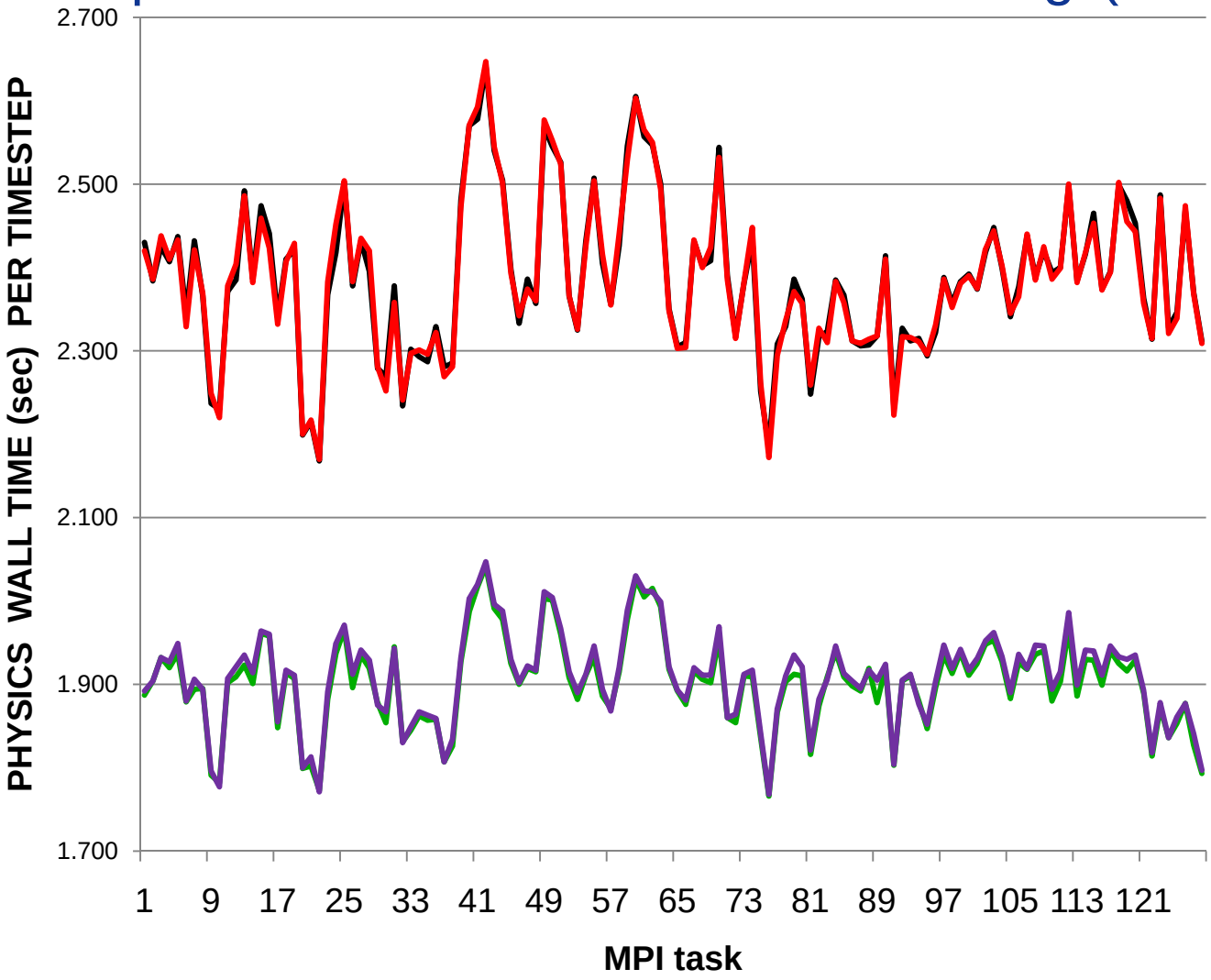
IFS Semi-Lagrangian Comms

- **Optimisation made to 4D-Var minimisation jobs (ifsmin)**
  - **Only first minimisation iteration requires constant width halo, i.e. ‘the blue points’**
  - **Remaining minimisation iterations (2 up to 70) use halo [procs,mask] info saved per time-step in first iteration, i.e. only ‘the red points’ are communicated**
  - **3 percent wall time improvement to 4D-Var ifsmin jobs**

Physics Computations

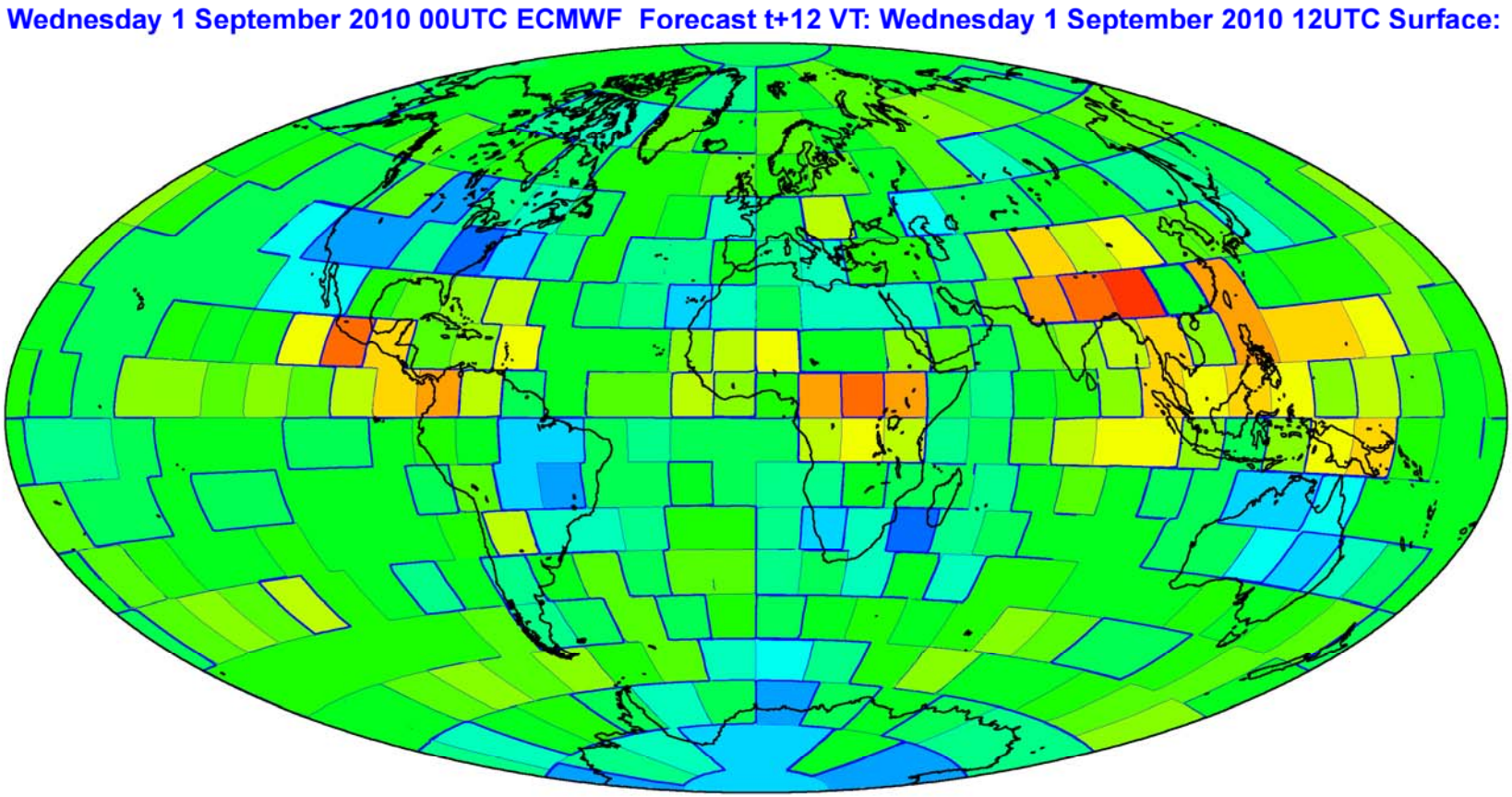
- **Take 34 percent of T1279L91 model wall time**
  - **Using 384 MPI tasks x 8 OpenMP threads = 3072 user threads**
- **About 3 percent is CPU work imbalance over MPI tasks**
- **Physics load balancing experiment**
  - **Global exchange of physics ‘cost’ info from earlier time step**
  - **Calculation of grid column ‘physics input’ comms schedule**
  - **Senders: pack data and post one or more MPI_SEND’s**
  - **Receivers: post one or more MPI_RECV’s and unpack data**
  - **Call ‘balanced’ physics**
  - **Receivers return ‘physics output’ back to senders…**
- **Other approaches to load balance computations?**

T1279 model 128 tasks : using 'previous' time step cost to drive load balancing (nradfr=6)



nstep	'previous'
0	-
1	-
2	1
3	2
4	3
5	4
6	**0**
7	5
8	7
9	8

T1279 384 MPI tasks: physics cost



Physics Load Balancing (conclusion)

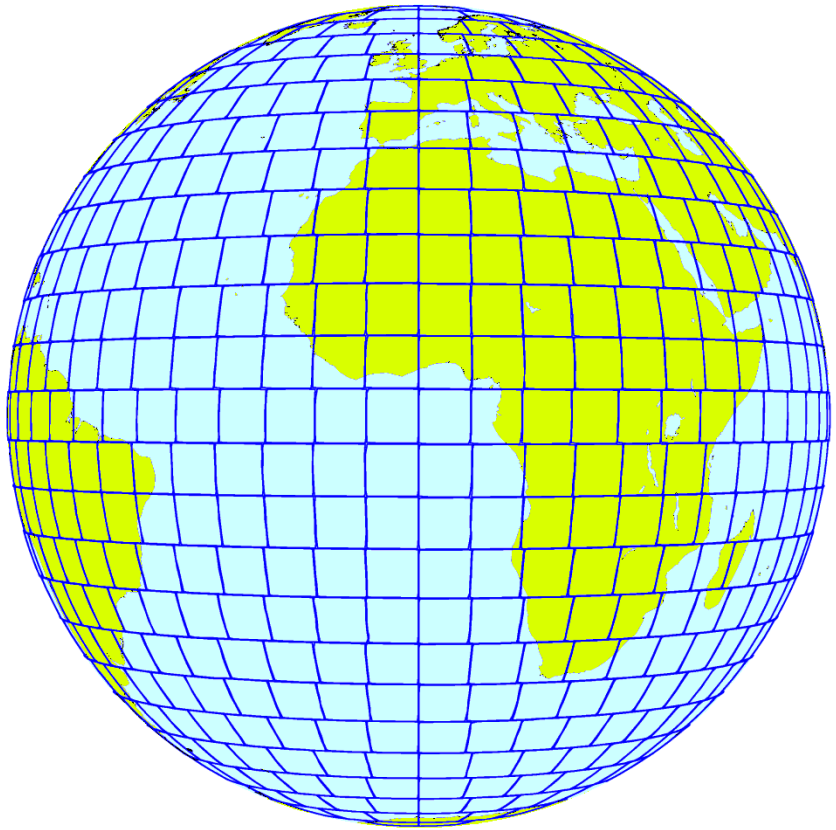
- **Overall performance of IFS model with load balanced physics was just faster (<1 percent) than using the original unbalanced physics**
- **Too much overhead in MPI message passing**
  - **mostly in sending grid column input and receiving back grid column output**
  - **Potential for optimisation using coarrays (F2008)**
- **Hard to hide load balancing code**
  - **the original code is already too complex with many arguments passed to ECMWF physics main routine (CALLPAR)**
- **Need significant performance gain before implementing such load balancing code**
  - **depends on more imbalance in future physics calculations**

Scaling issues: today

- **Static Load Imbalance**
  - **per MPI task, per OpenMP thread**
- **Dynamic Load Imbalance**
  - **e.g. physics computations, semi-Lagrangian comms**
- **Jitter**
- **MPI Comms Latency, Topology**
- **OpenMP overheads, NUMA**
- **Input/Output**
- **Shell scripts**

Scaling to 100K - 1M threads ?

- **Next 5 to 10 years ?**
- **Can this still be done with MPI + OpenMP ?**
- **Partitioned global address space (PGAS) languages?**
- **Fortran 2008 coarrays**
- **Jitter free systems**
- **Need comms to speed up with the increase in cores**
- **Overlap compute and comms?**
- **Tools (debuggers, profilers)**
  - **That work reliably and fast at high core counts**
  - **That work with large applications**



Questions

T159L91 model using 2048 user threads: Are we running out of parallelism?

	512 tasks x 4 threads	Comments
Grid Point computations	Model grid: 35,718 grid points each task: 70-71 each thread: 17-18 SL halo: >1000 grid points (1 hr time-step) Radiation grid (T63): 6114 grid points each task: 11-12 grid points each thread: 3-4 grid points	1 nproma chunk per thread, so no OpenMP load balancing Halo is ~14 times size of grid points owned by each task
Fourier transforms	160 lats x 91 levels = 14,560 Partitioning 32 x 16 each task: 5 lats x 5-6 levels each thread: 1-2 lats x 5-6 levels	Poor static distribution
Legendre transforms	160 waves x 91 levels Partitioning 32 x 16 each task: 5 waves x 5-6 levels each thread: 1-2 waves x 5-6 levels	Poor static distribution

Power6: SpeedUp and Efficiency
T799L91 model, 2 day forecast (CY36R4)

parallel	serial
475124	65

User Threads	Actual Wall Time	Calculated Wall Time	Calculated SpeedUp	Calculated Efficiency %
256	1925.9	1921	247	96.4
384	1286.2	1302	369	96.2
512	1001.3	993	475	92.7
768	685.5	683	693	90.3
1024	521.2	529	912	89.0
1280	448.9	436	1059	82.7
1536	379.2	374	1253	81.6
1920	300.8	312	1580	82.3
1		475189	**10 day forecast ~ 45 min**	

ECMWF