CRESTA

A PGAS Implementation of the ECMWF Integrated Forecasting System (IFS) NWP Model

George.Mozdzynski@ecmwf.int

Acknowledgements

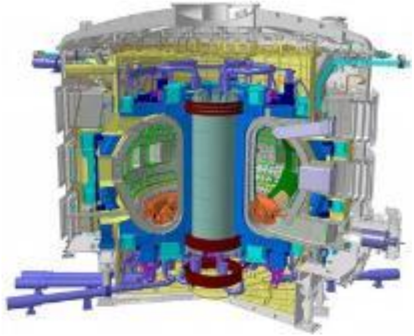
Mats Hamrud	ECMWF
Nils Wedi	ECMWF
Willem Deconinck	ECMWF
Jens Doleschal	Technische Universität Dresden
Harvey Richardson	Cray UK

And my other partners in the CRESTA Project

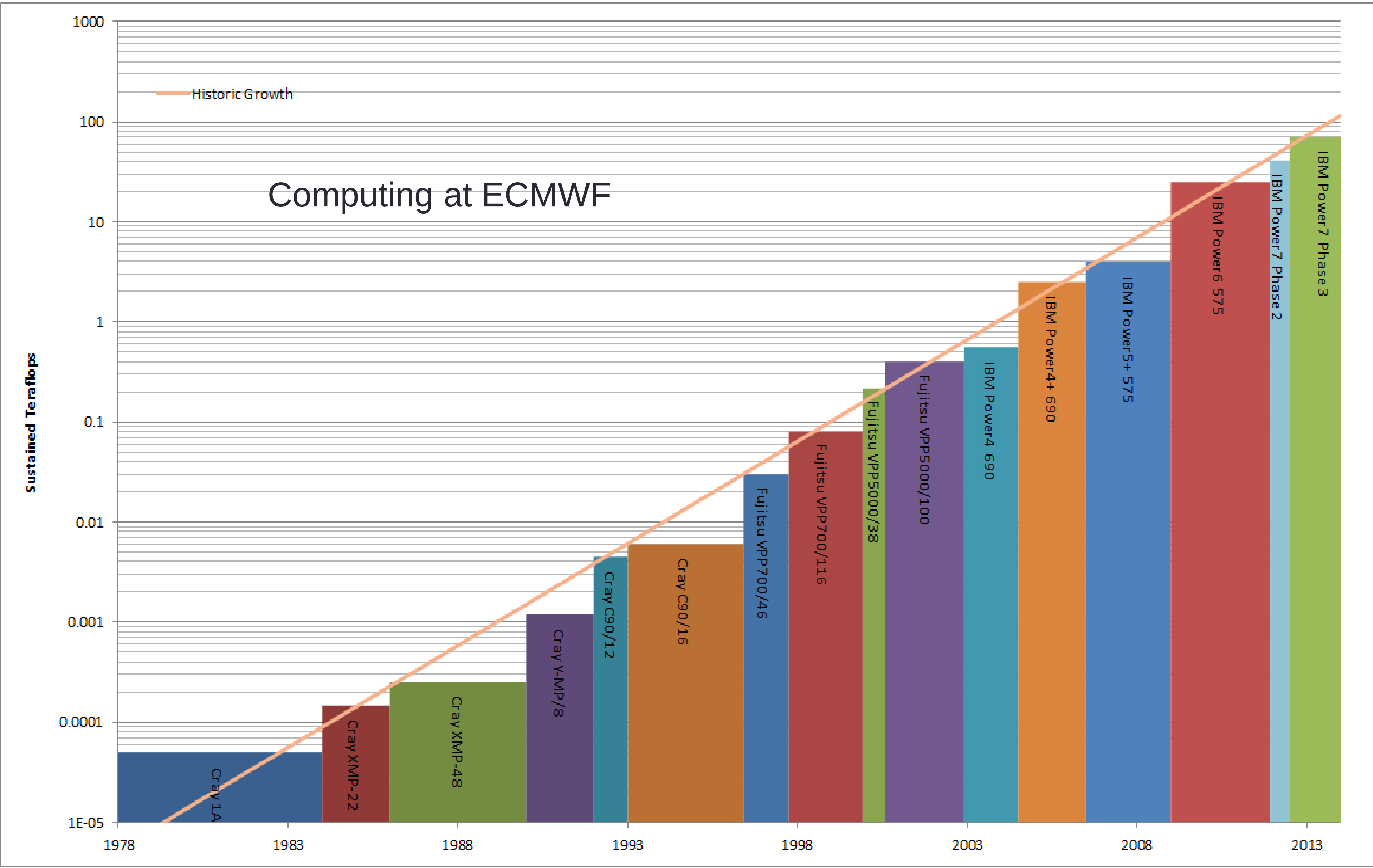
The CRESTA project has received funding from the EU Seventh Framework Programme (ICT-2011.9.13)

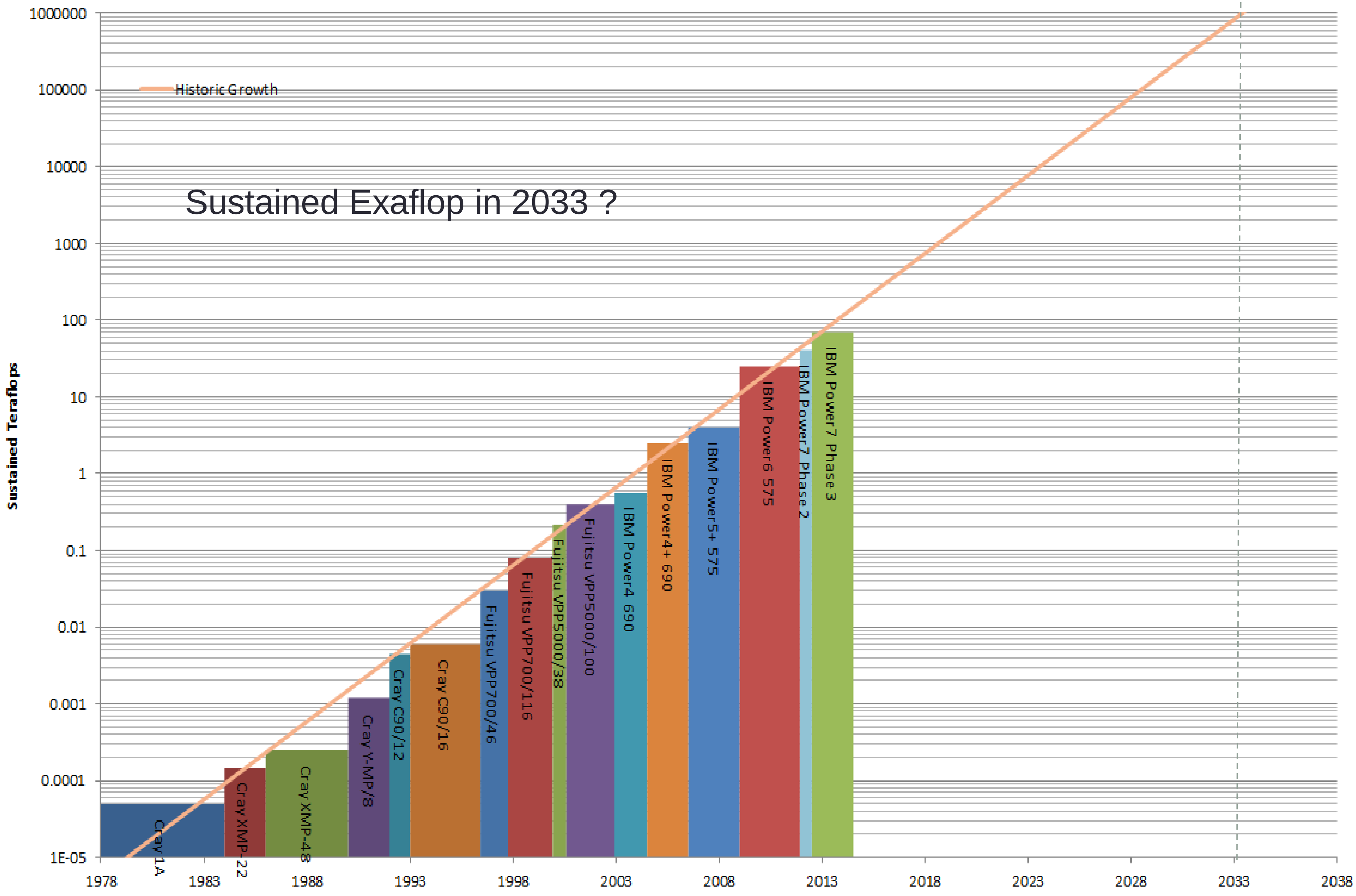
What is CRESTA - see http://cresta-project.eu/

- Collaborative Research into Exascale Systemware, Tools and Applications
- EU funded project, 3 years (started Oct 2011), ~ 50 scientists
- Six co-design vehicles (aka applications)
  - ELMFIRE (CSC, ABO,UEDIN) - fusion plasma
  - GROMACS (KTH) - molecular dynamics
  - HEMELB (UCL) - biomedical
  - IFS (ECMWF) - weather
  - NEK5000 (KTH) & OPENFOAM (USTUTT, UEDIN) - comp. fluid dynamics



- Two tool suppliers
  - ALLINEA (ddt : debugger) & TUD (vampir : performance analysis)
- Technology and system supplier – CRAY UK
- Many Others (mostly universities)
  - ABO, CRSA, CSC, DLR, JYU, KTH, UCL, UEDIN-EPCC, USTUTT-HRLS


Computing at ECMWF
Historic Growth
Sustained Teraflops
1000
100
10
1
0.1
0.01
0.001
0.0001
1E-05
1978
1983
1988
1993
1998
2003
2008
2013
Cray 1A
Cray XMP-22
Cray XMP-48
Cray Y-MP/8
Cray C90/12
Cray C90/16
Fujitsu VPP700/46
Fujitsu VPP700/116
Fujitsu VPP5000/38
Fujitsu VPP5000/100
IBM Power4 690
IBM Power4+ 690
IBM Power5+ 575
IBM Power6 575
IBM Power7 Phase 2
IBM Power7 Phase 3


Historic Growth
Sustained Exaflop in 2033 ?
Sustained Teraflops
1000000
100000
10000
1000
100
10
1
0.1
0.01
0.001
0.0001
1E-05
1978
1983
1988
1993
1998
2003
2008
2013
2018
2023
2028
2033
2038
Cray 1A
Cray XMP-22
Cray XMP-48
Cray Y-MP/8
Cray C90/12
Cray C90/16
Fujitsu VPP700/46
Fujitsu VPP700/116
Fujitsu VPP5000/38
Fujitsu VPP5000/100
IBM Power4 690
IBM Power4+ 690
IBM Power5+ 575
IBM Power6 575
IBM Power7 Phase 2
IBM Power7 Phase 3

IFS model: current and future model resolutions

IFS model resolution	Envisaged Operational Implementation	Grid point spacing (km)	Time-step (seconds)	Estimated number of cores[1]
T1279 H[2]	2013 (L137)	16	600	2K
T2047 H	2014-2015	10	450	6K
T3999 NH[3]	2023-2024	5	240	80K
T7999 NH	2031-2032	2.5	30-120	1-4M

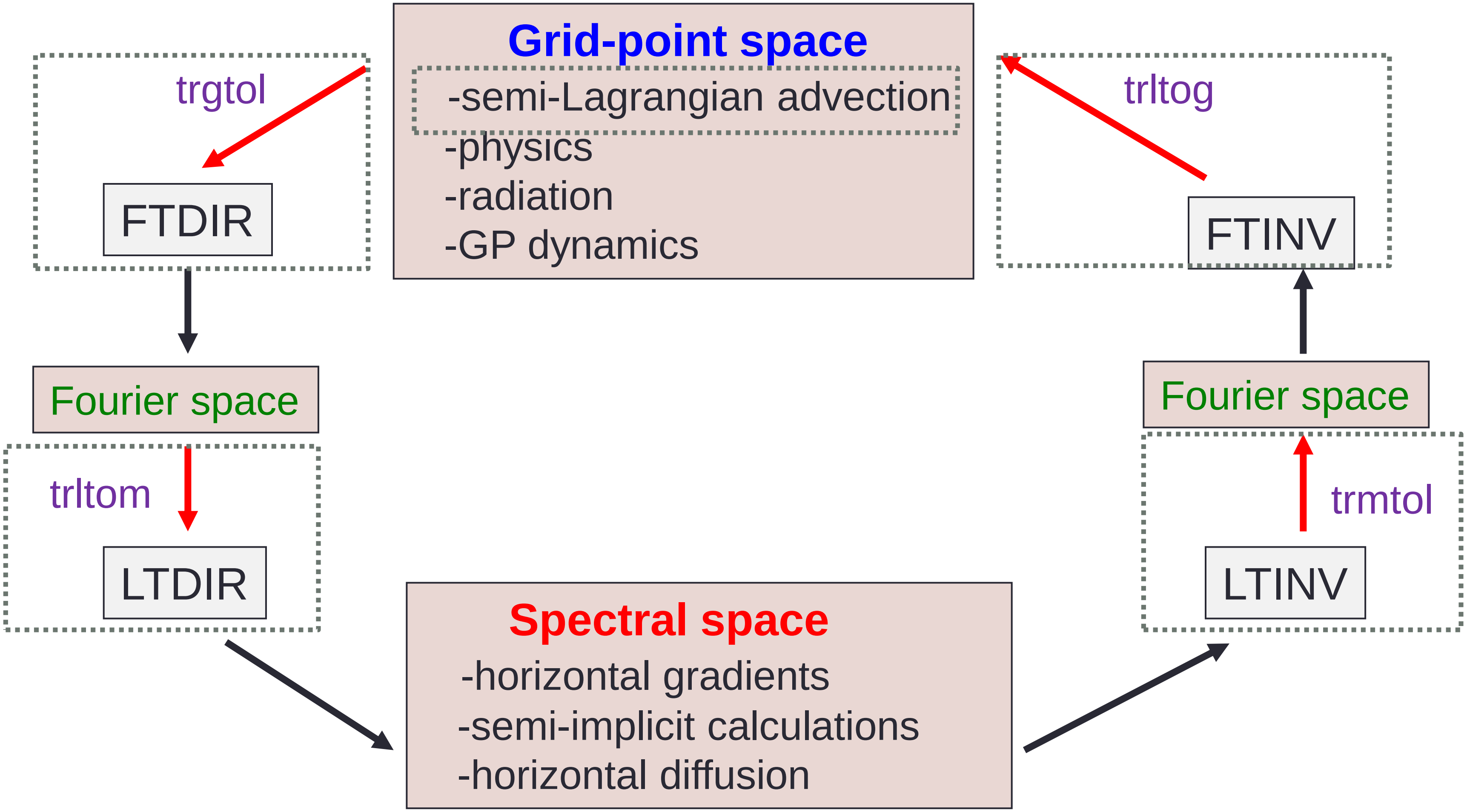
1 – a gross estimate for the number of 'IBM Power7' equivalent cores needed to achieve a 10 day model forecast in under 1 hour (~240 FD/D), system size would normally be ~10 times this number.
2 – Hydrostatic Dynamics
3 – Non-Hydrostatic Dynamics

IFS PGAS Optimizations for ExaScale & Co-design

- IFS PGAS optimizations in the CRESTA project
  - Involve use of Fortran2008 coarrays (CAF)
  - Used within context of OpenMP parallel regions
- Overlap Legendre transforms with associated transpositions
- Overlap Fourier transforms with associated transpositions
- Rework semi-Lagrangian communications
  - To substantially reduce communicated halo data
  - To overlap halo communications with SL interpolations
- CAF co-design team
  - caf-co-design@cresta-project.eu
  - ECMWF – optimize IFS as described above
  - CRAY – optimize DMAPP to be thread safe
  - TUD – visualize CAF operations in IFS with vampir
  - ALLINEA – debug IFS at scale with ddt (MPI/OMP/CAF)

george.mozdzynski@ecmwf.int
mats.hamrud@ecmwf.int
willem.deconinck@ecmwf.int
harveyr@cray.com
michs@kth.se
tobias.hilbrich@tu-dresden.de
kostas@ihs.uni-stuttgart.de
m.bull@epcc.ed.ac.uk
jens.doleschal@tu-dresden.de
xaguilar@pdc.kth.se
david@allinea.com
jeremy@epcc.ed.ac.uk

IFS PGAS optimizations for [Tera,Peta,Exa]scale



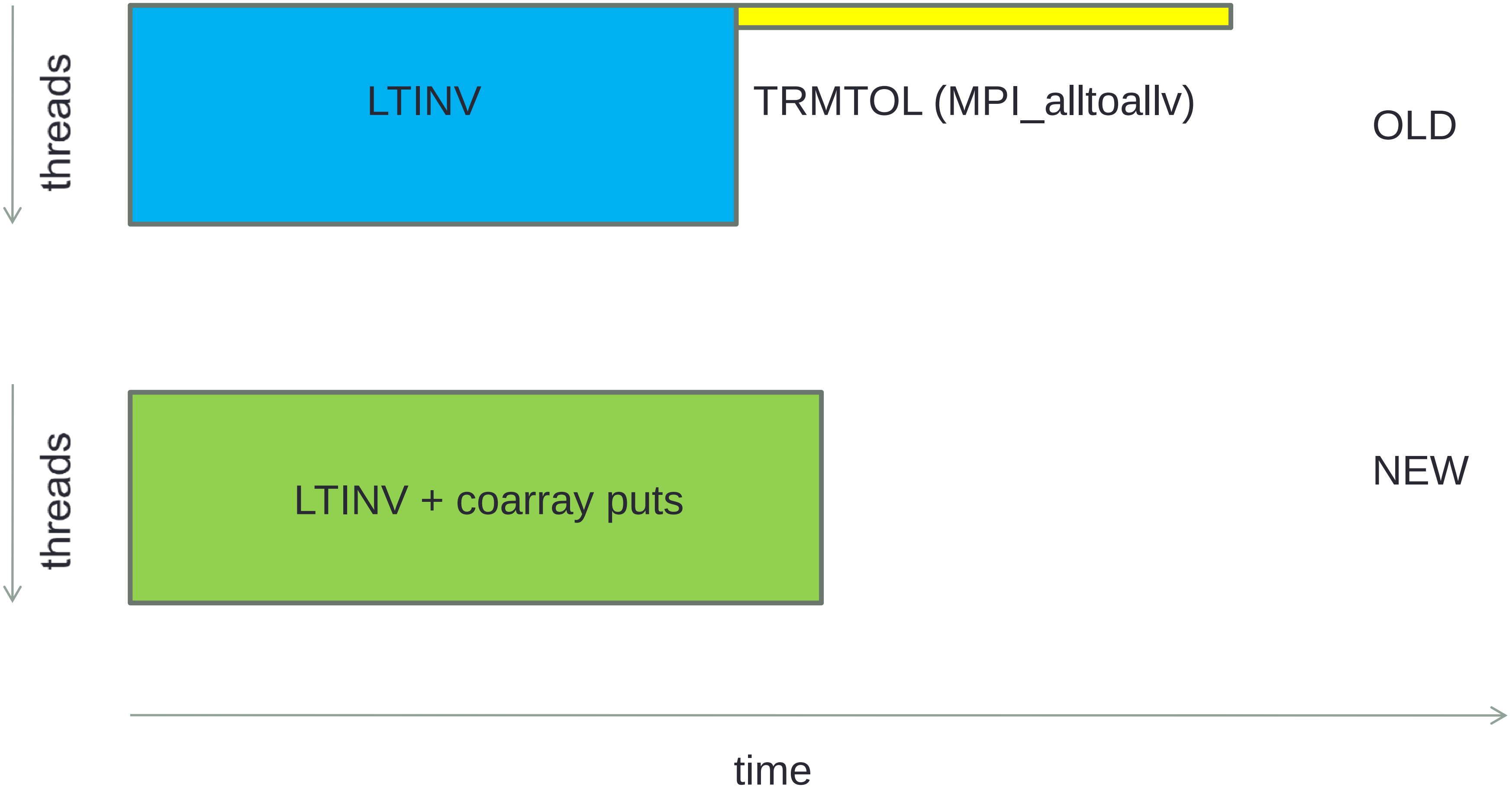
Grid-point space

-semi-Lagrangian advection
-physics
-radiation
-GP dynamics

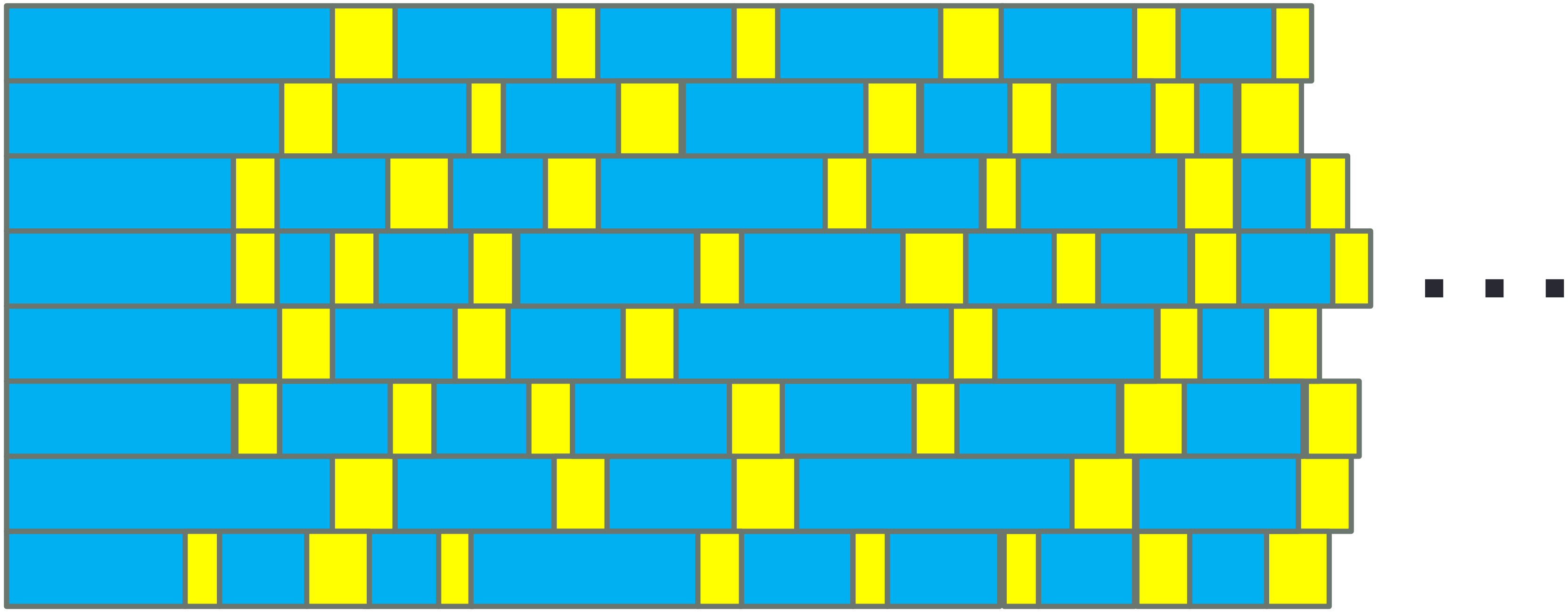
Spectral space

-horizontal gradients
-semi-implicit calculations
-horizontal diffusion

Overlap Legendre transforms with associated transpositions



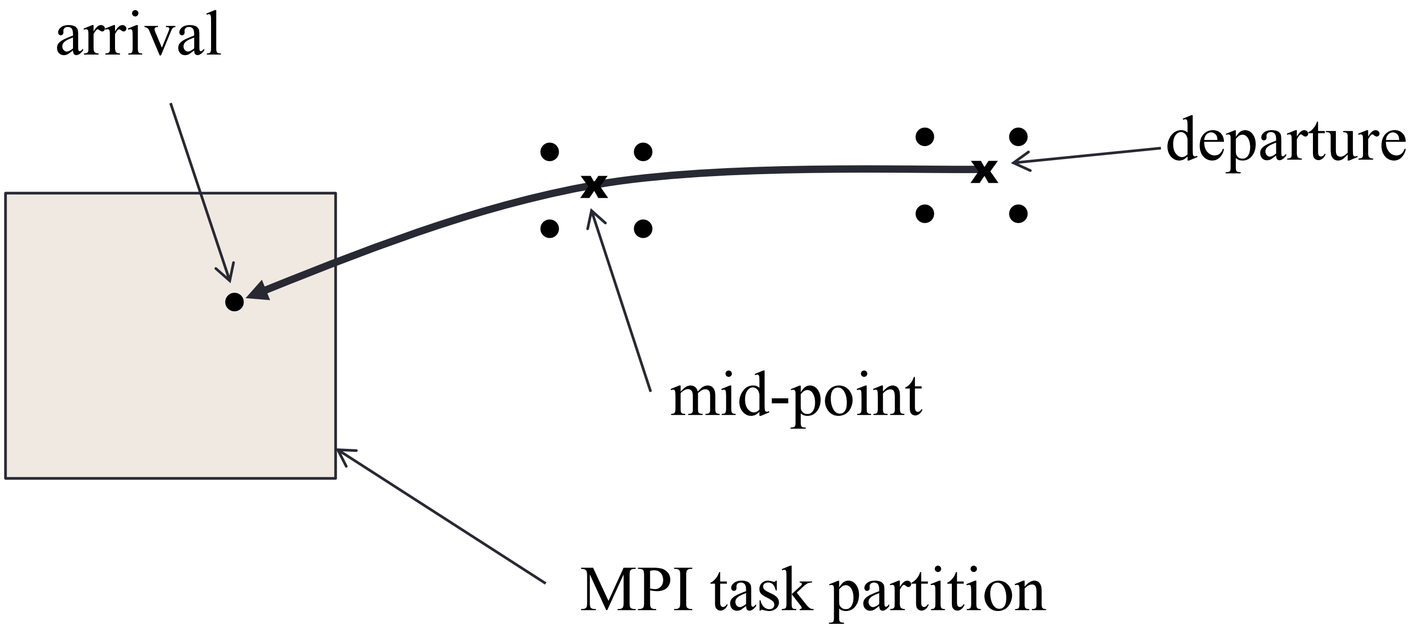
Overlap Legendre transforms with associated transpositions/3 (LTINV + coarray puts)



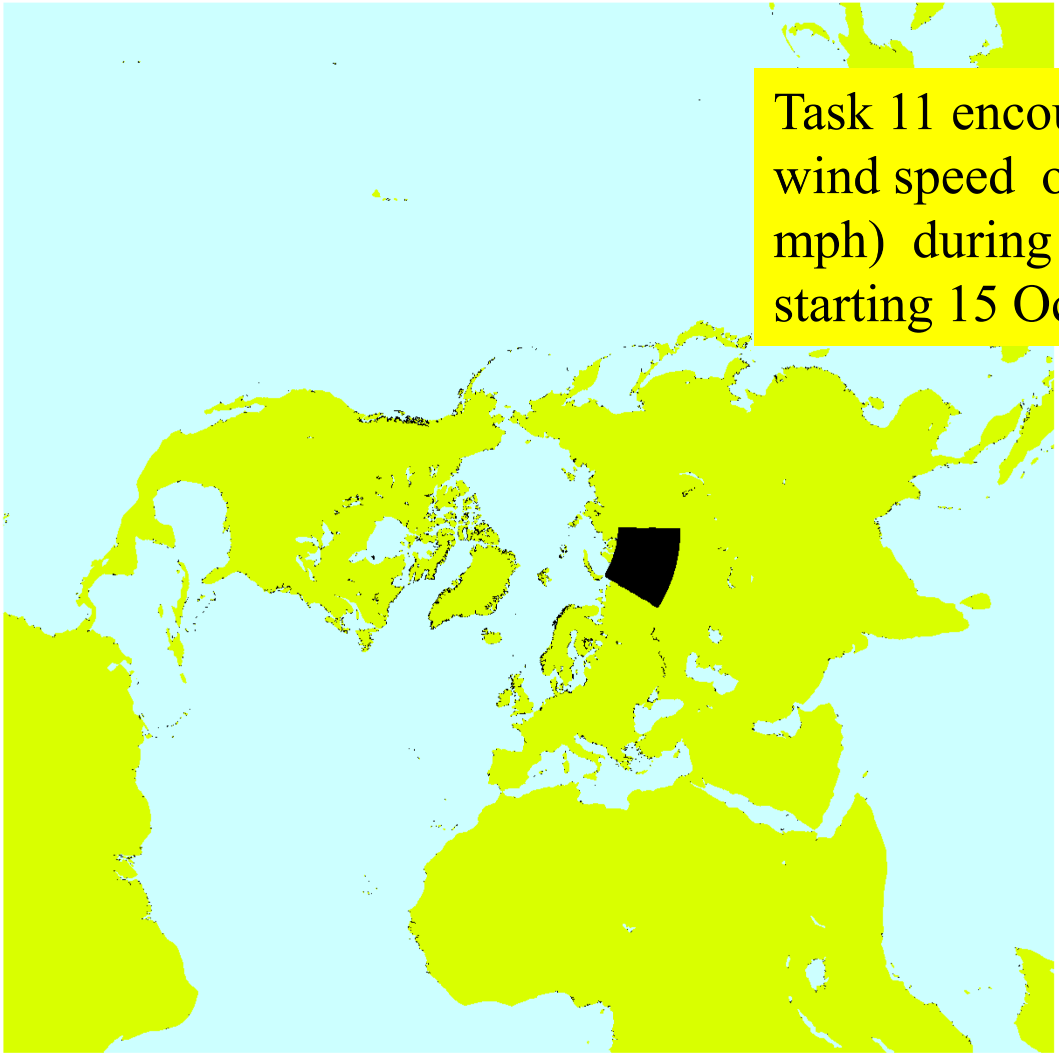
Expectation is that compute (LTINV-blue) and communication (coarray puts-yellow) overlap in time. We can now see this with an extension to vampir developed in CRESTA

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

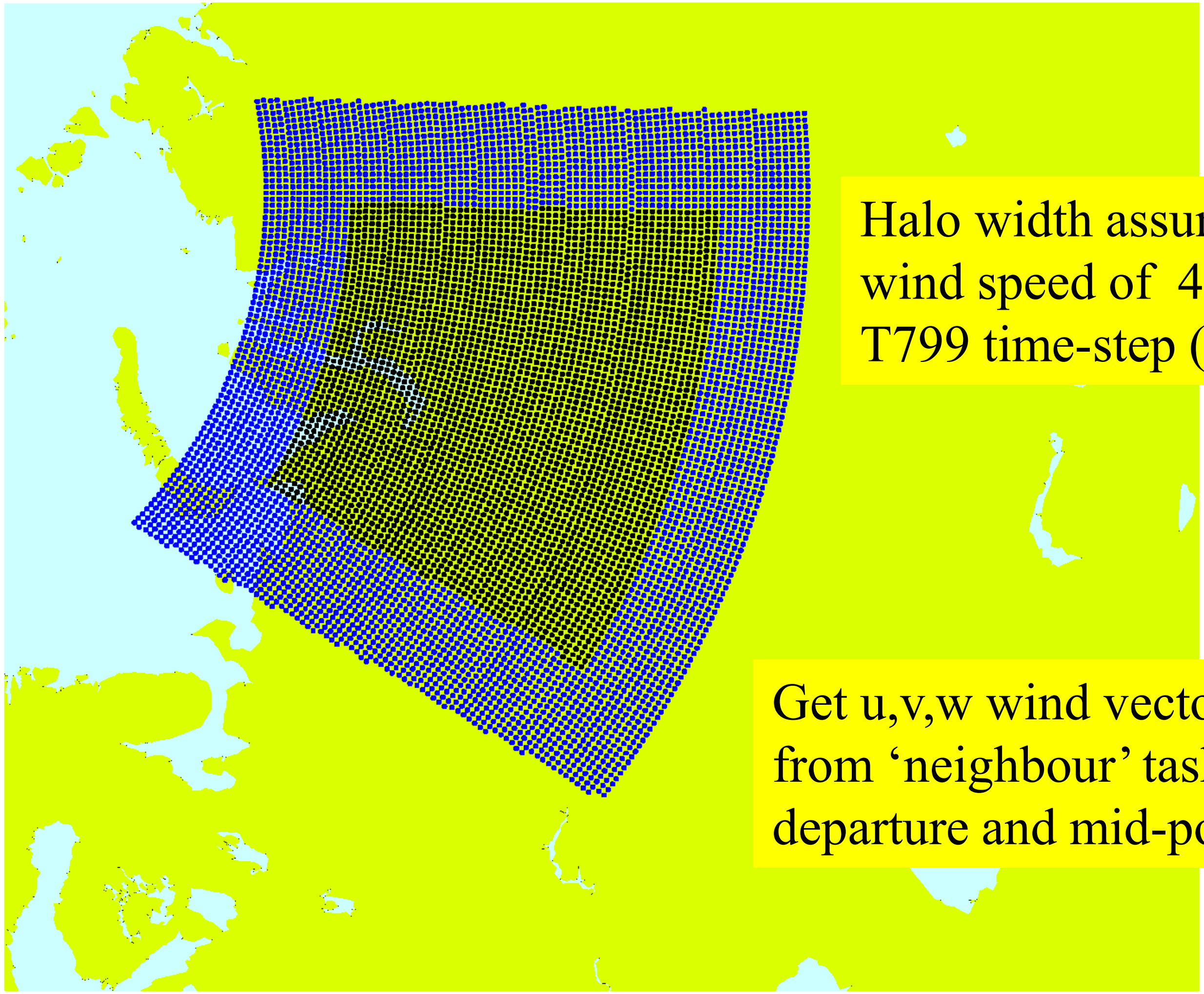


Semi-Lagrangian Transport: T799 model, 256 tasks



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

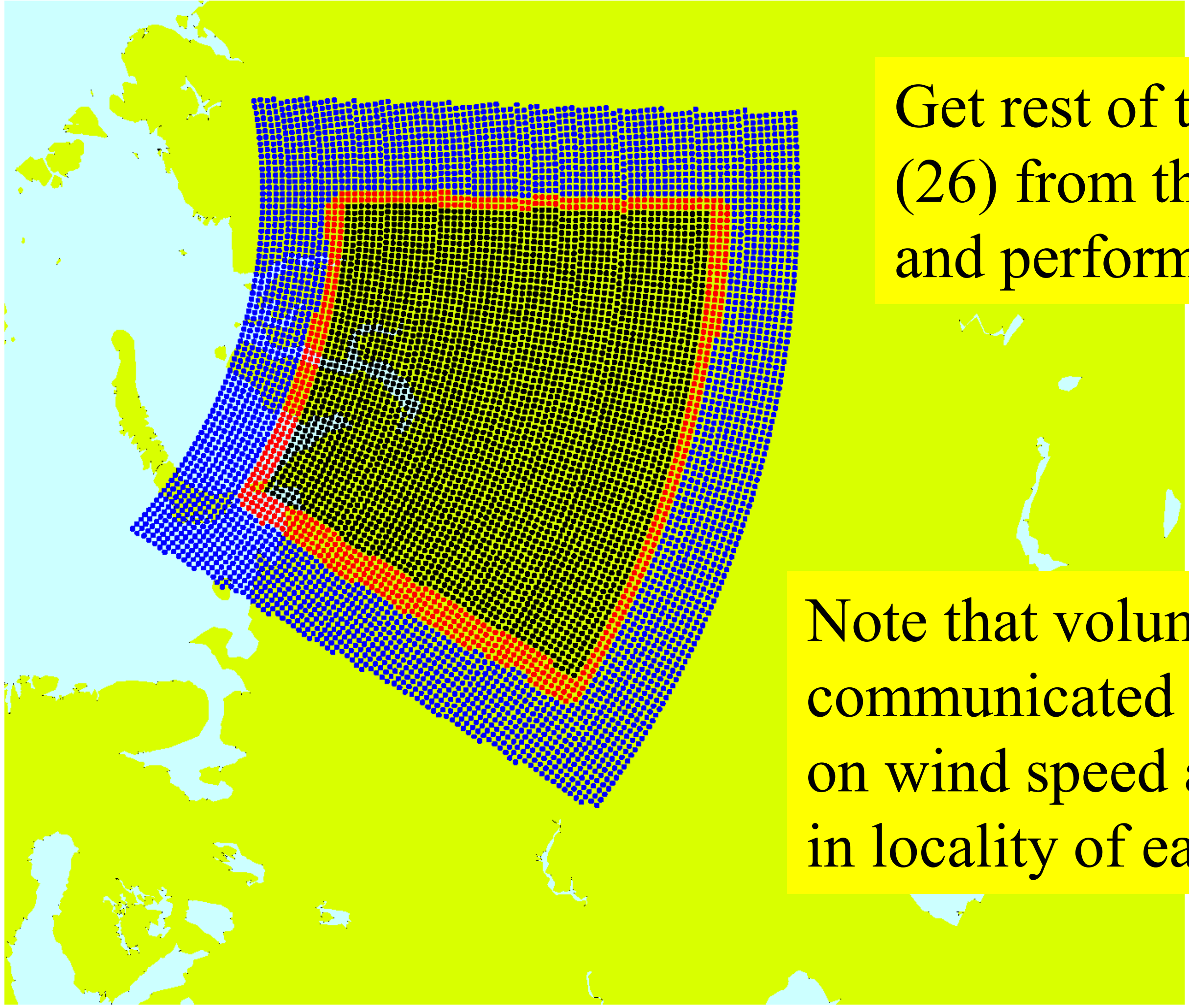
blue: halo area



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

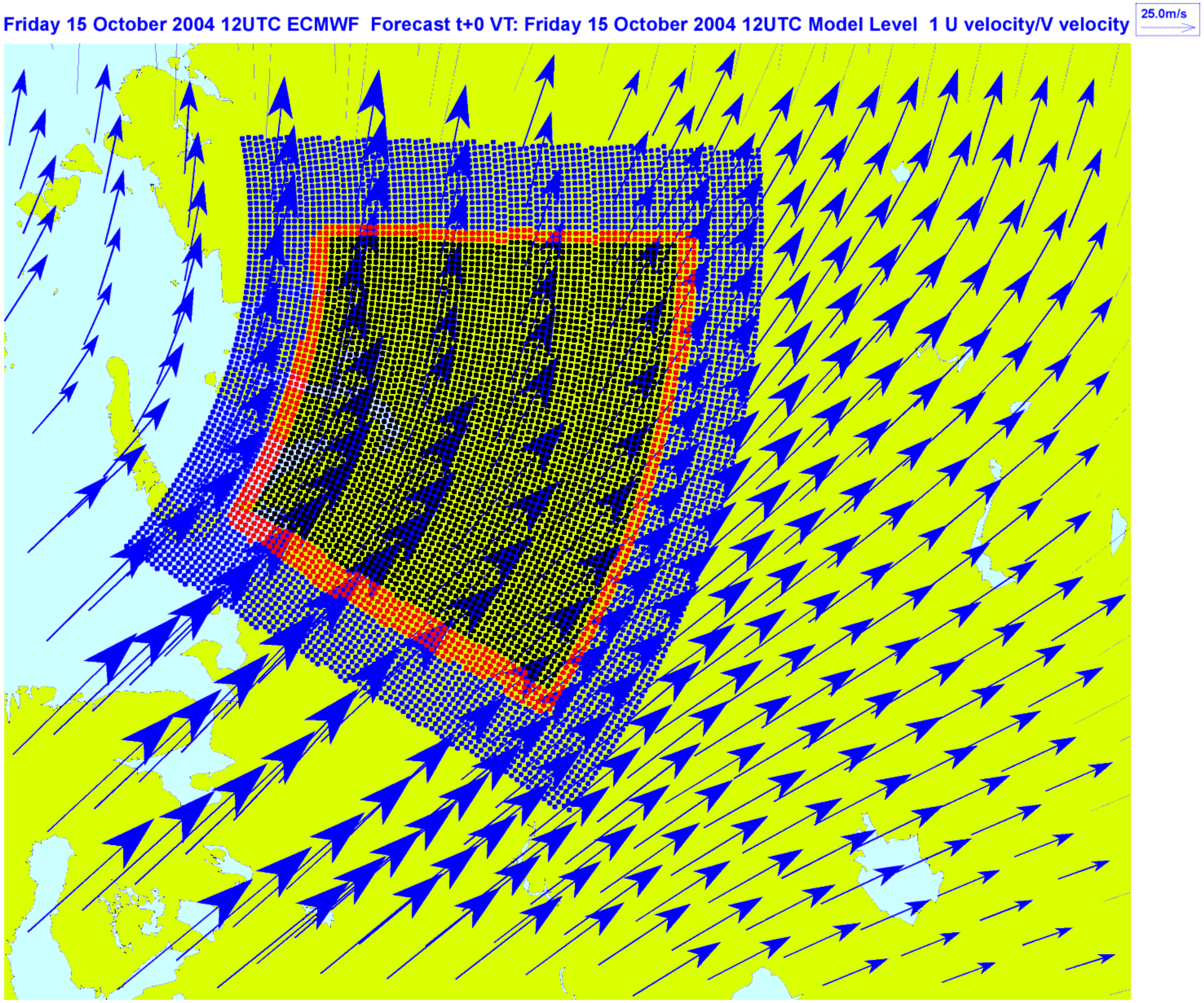
red: halo points actually used



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

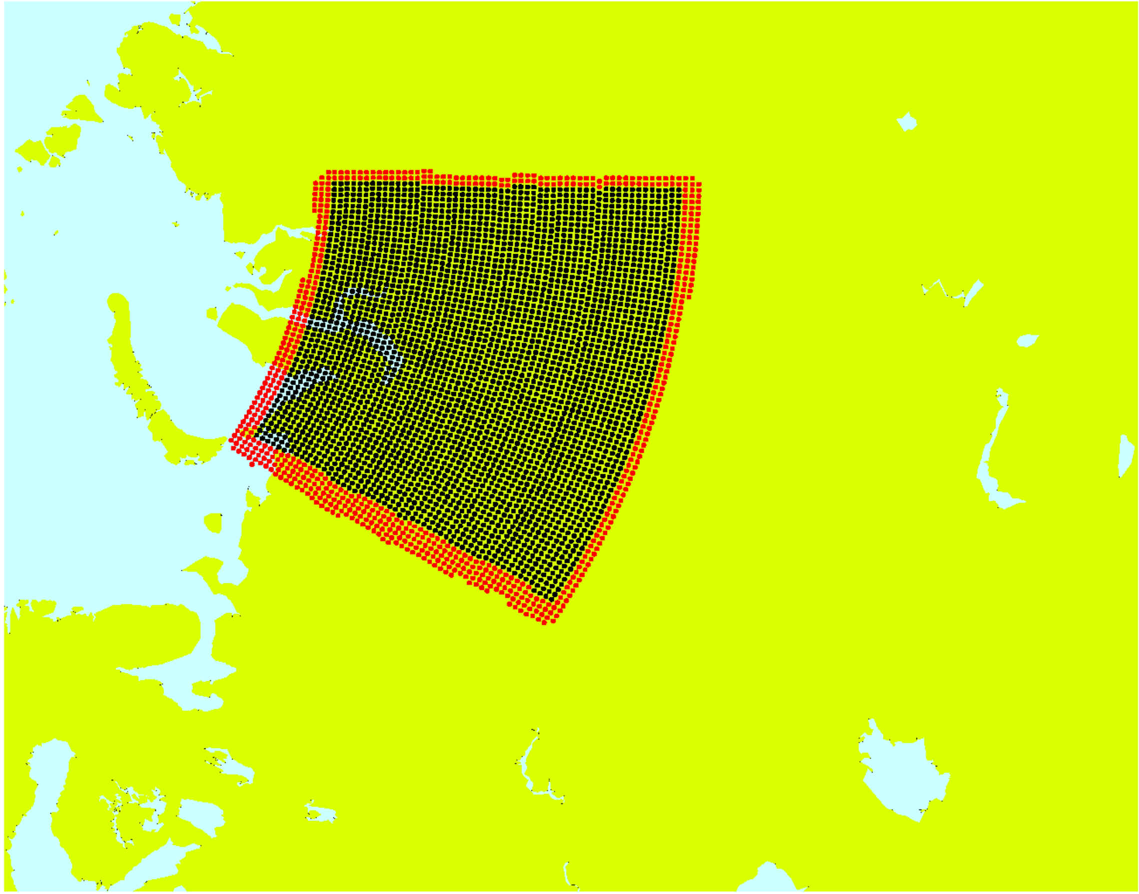
Note that volume of halo data communicated is dependent on wind speed and direction in locality of each task

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

Semi-Lagrangian – coarray implementation

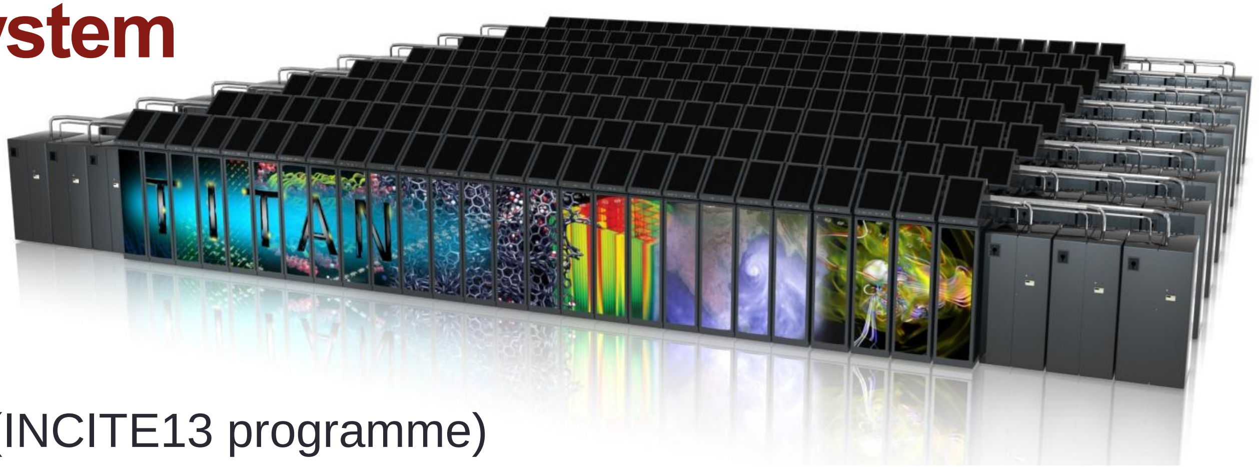
red: only the halo points that are used are communicated



Note no more blue area (max wind halo) and associated overhead.

Also, halo coarray transfers take place in same OpenMP loop as the interpolations.

ORNL's "Titan" System

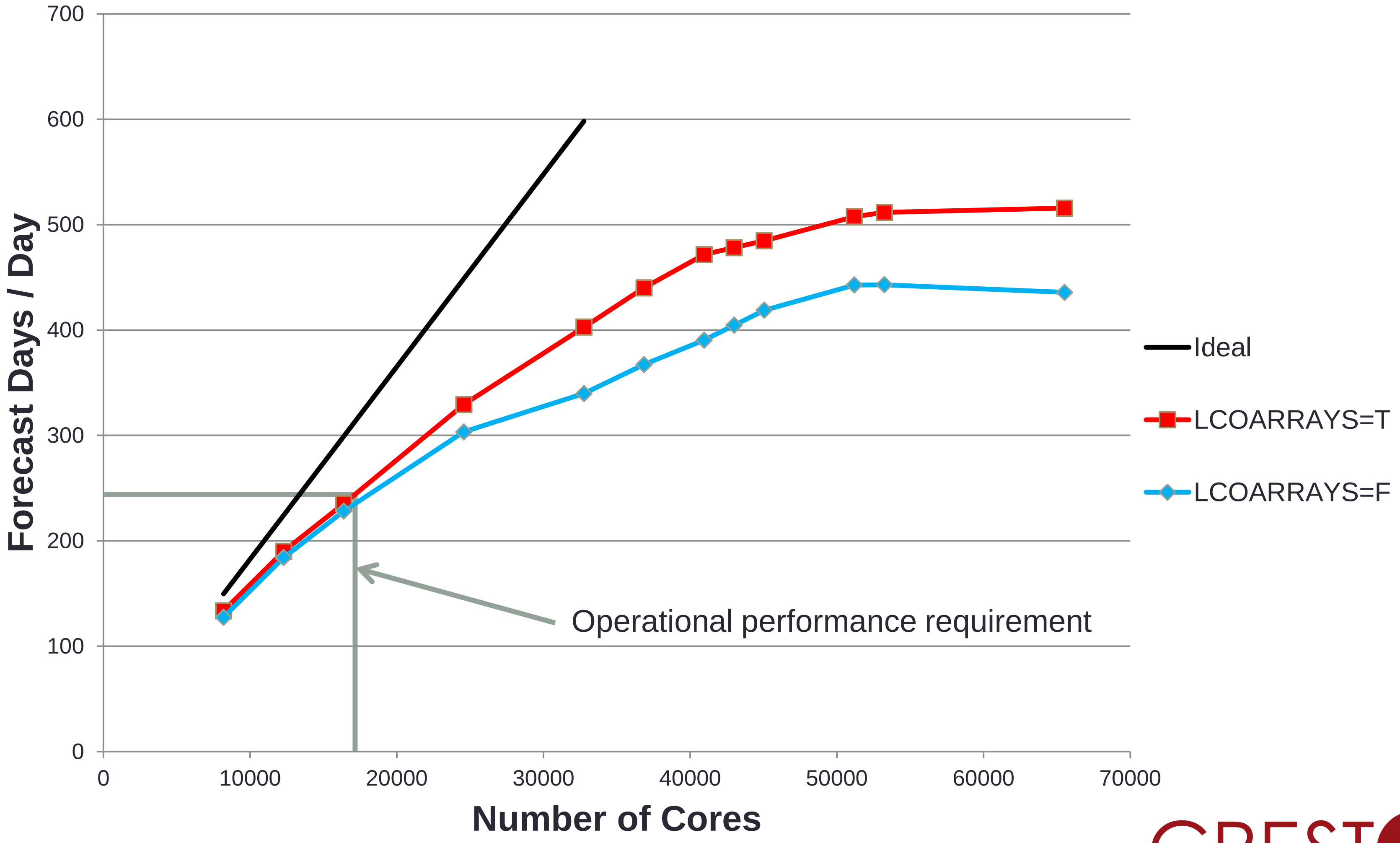


- #1 in Nov 2012 Top500 list
- CRESTA awarded access (INCITE13 programme)
- 18X peak perf. of ECMWF's P7 clusters (C2A+C2B=1.5 Petaflops)
- Upgrade of Jaguar from Cray XT5 to XK6
- Cray Linux Environment operating system
- Gemini interconnect
  - 3-D Torus
  - Globally addressable memory
- AMD Interlagos cores (16 cores per node)
- New accelerated node design using NVIDIA K20 "Kepler" multi-core accelerators
- 600 TB DDR3 mem. + 88 TB GDDR5 mem

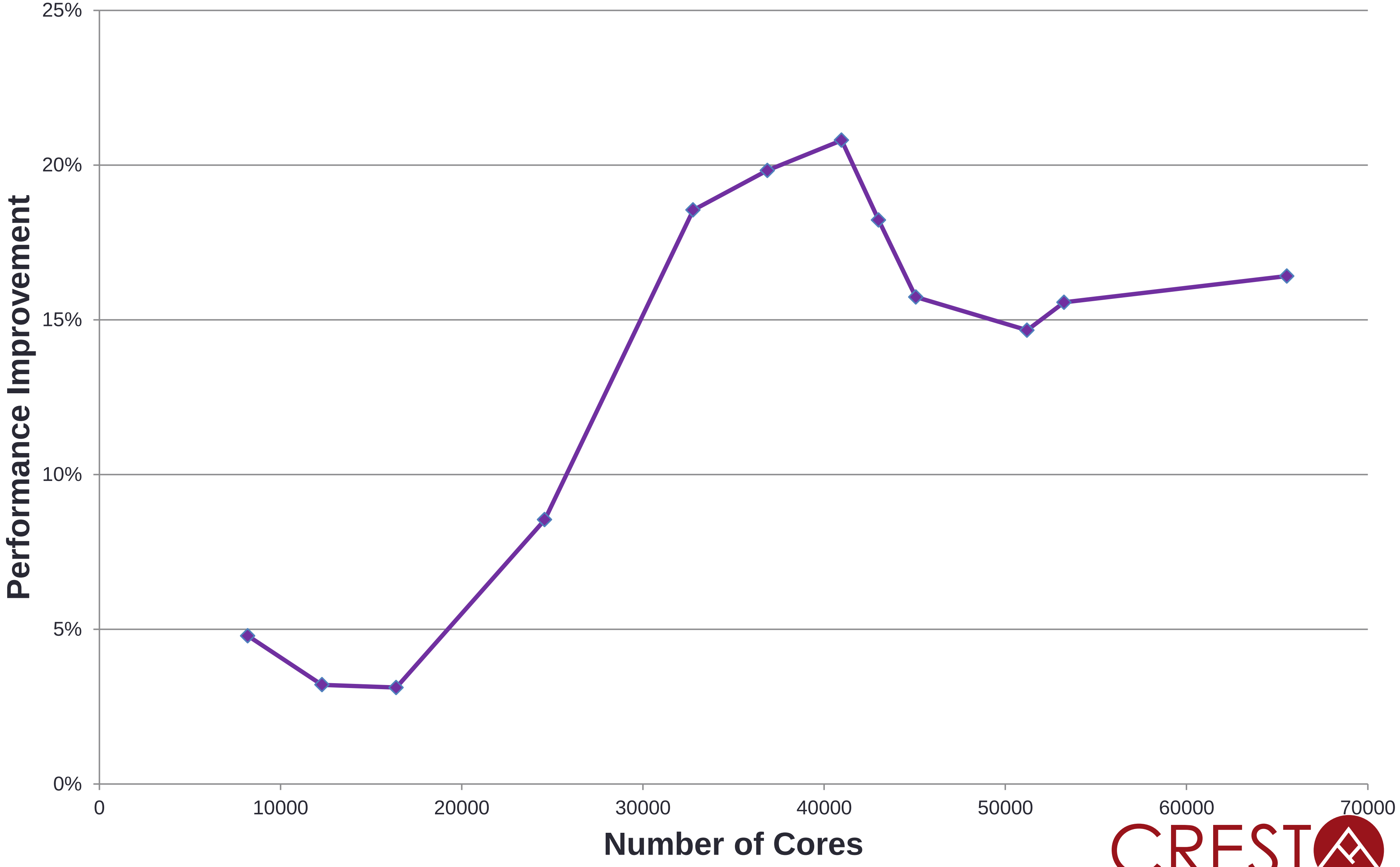
Titan Specs	
Compute Nodes	18,688
Login & I/O Nodes	512
Memory per node	32 GB + 6 GB
# of NVIDIA K20 "Kepler" processors	14,592
Total System Memory	688 TB
Total System Peak Performance	27 Petaflops

Source (edited): *James J. Hack, Director,* Oak Ridge National Laboratory

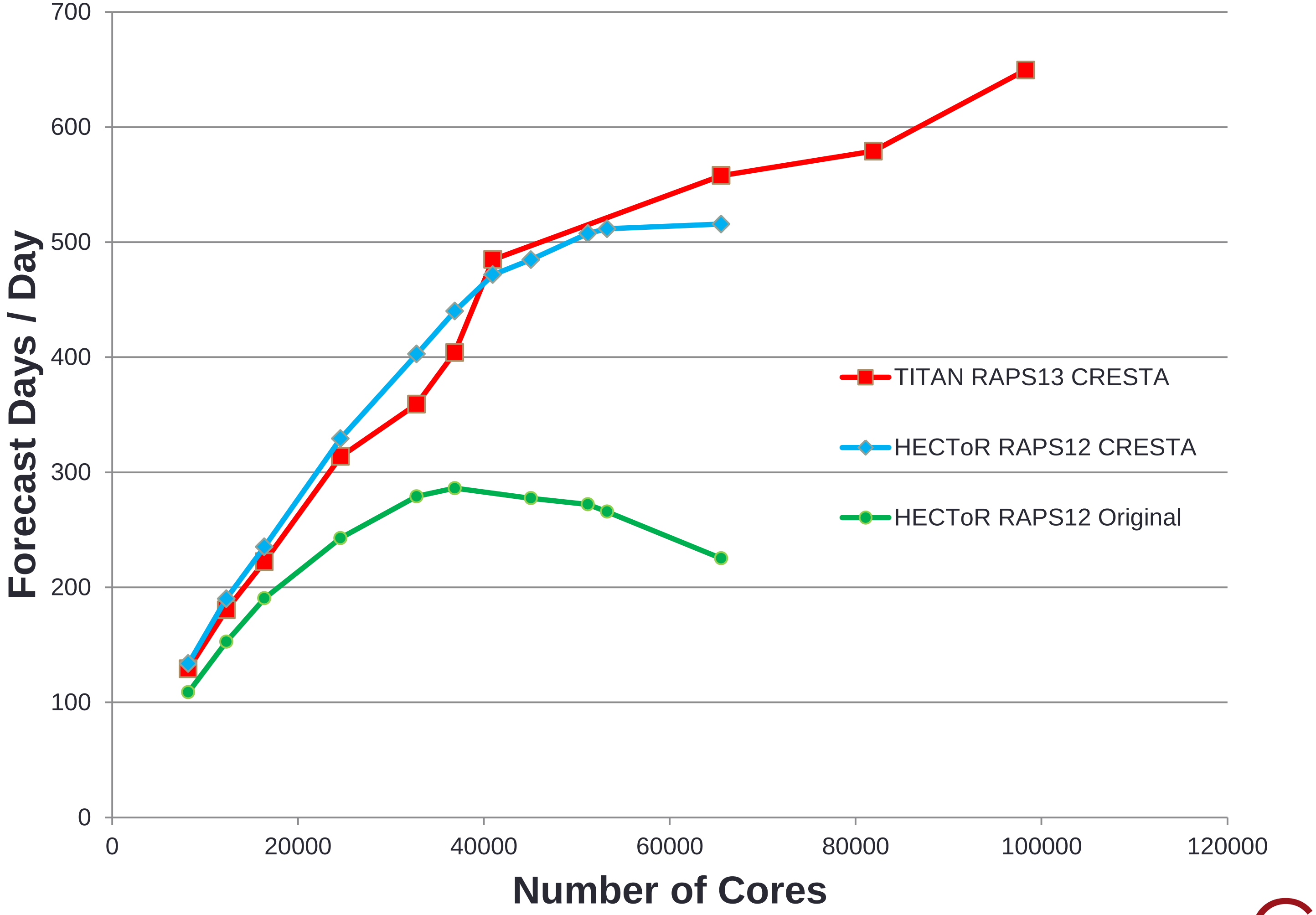
T2047L137 model performance on HECToR (CRAY XE6)
RAPS12 IFS (CY37R3), cce=8.0.6 -hflex_mp=intolerant



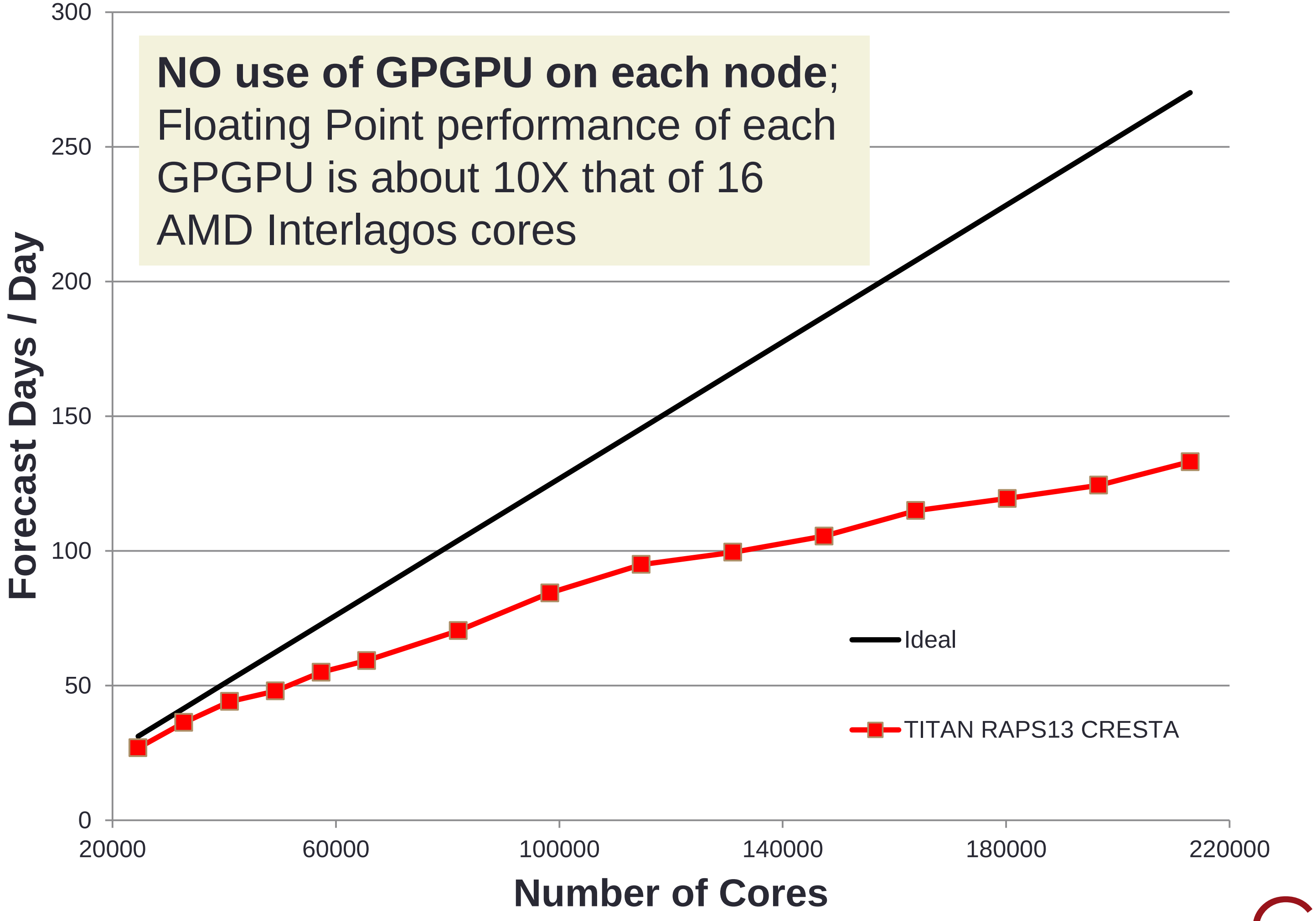
T2047L137 IFS model performance improvement by using Fortran2008 coarrays on HECToR (CRAY XE6)



T2047L137 IFS forecast model performance
RAPS12 (CY37R3, on HECToR), RAPS13 (CY38R2, on TITAN)



IFS T3999L137 hydrostatic forecast model performance on TITAN RAPS13 IFS (CY38R2), cce=8.1.5, NRADRES=2047, NRADFR=1



NO use of GPGPU on each node; Floating Point performance of each GPGPU is about 10X that of 16 AMD Interlagos cores

LTINV recoding

COMPUTE

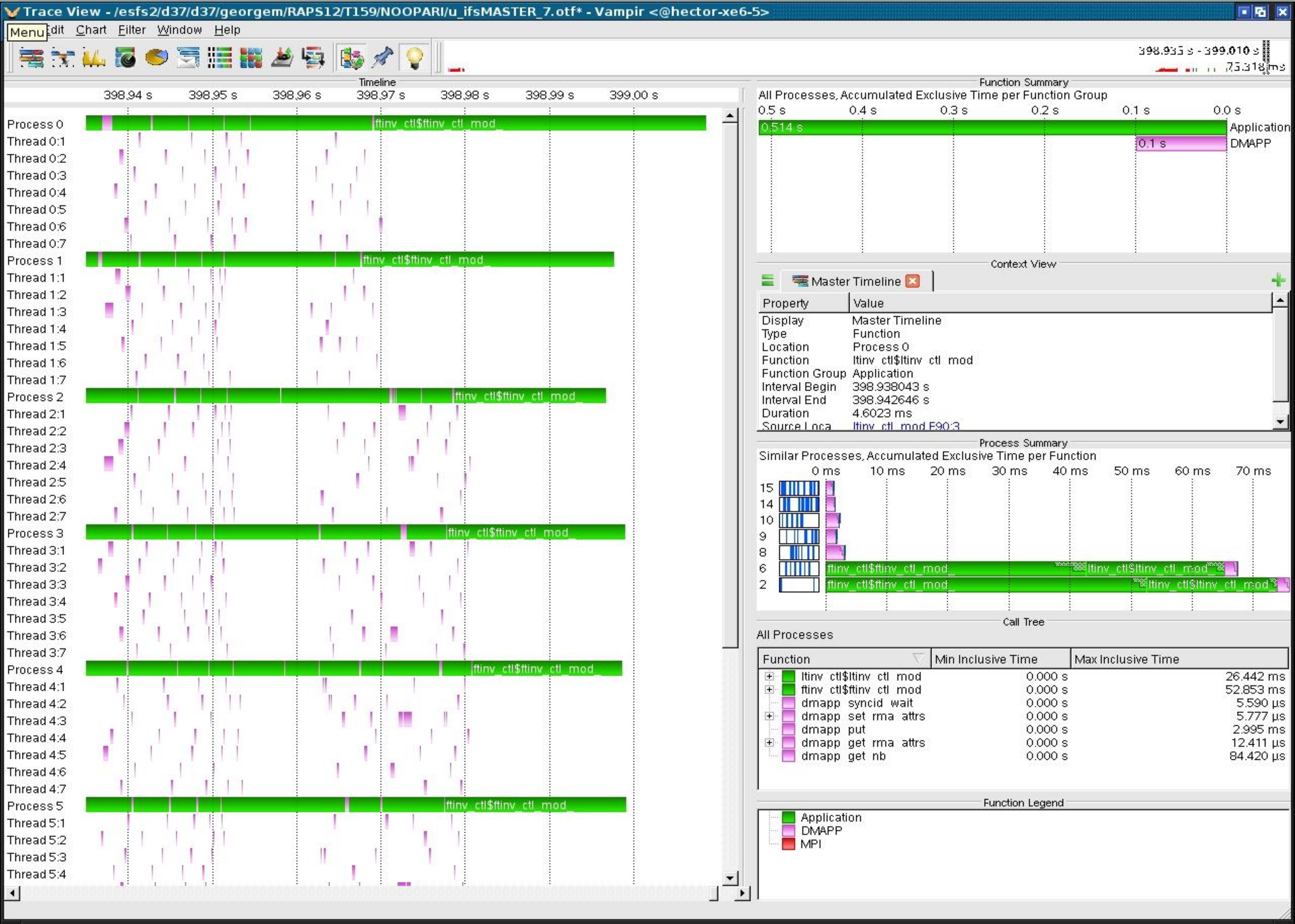
COMMUNICATION

ORIGINAL code

```
!$OMP PARALLEL DO SCHEDULE(DYNAMIC,1) PRIVATE(JM,IM)
DO JM=1,D%NUMP
  IM = D%MYMS(JM)
  CALL LTINV(IM,JM,KF_OUT_LT,KF_UV,KF_SCALARS,KF_SCDERS,ILEI2,IDIM1,&
   & PSPVOR,PSPDIV,PSPSCALAR ,&
   & PSPSC3A,PSPSC3B,PSPSC2 , &
   & KFLDPTRUV,KFLDPTRSC,FSPGL_PROC)
ENDDO
!$OMP END PARALLEL DO
DO J=1,NPRTRW
  ILENS(J) = D%NLTSFTB(J)*IFIELD
  IOFFS(J) = D%NSTAGT0B(J)*IFIELD
  ILENR(J) = D%NLTSGTB(J)*IFIELD
  IOFFR(J) = D%NSTAGT0B(D%MSTABF(J))*IFIELD
ENDDO
CALL MPL_ALLTOALLV(PSENDBUF=FOUBUF_IN,KSENDCOUNTS=ILENS,&
 & PRECVBUF=FOUBUF,KRECVCOUNTS=ILENR,&
 & KSENDDISPL=IOFFS,KRECVDISPL=IOFFR,&
 & KCOMM=MPL_ALL_MS_COMM,CDSTRING='TRMTOL:')
```

NEW code

```
!$OMP PARALLEL DO SCHEDULE(DYNAMIC,1) PRIVATE(JM,IM,JW,IPE,ILEN,ILENS,IOFFS,IOFFR)
DO JM=1,D%NUMP
  IM = D%MYMS(JM)
  CALL LTINV(IM,JM,KF_OUT_LT,KF_UV,KF_SCALARS,KF_SCDERS,ILEI2,IDIM1,&
    & PSPVOR,PSPDIV,PSPSCALAR ,&
    & PSPSC3A,PSPSC3B,PSPSC2 , &
    & KFLDPTRUV,KFLDPTRSC,FSPGL_PROC)
  DO JW=1,NPRTRW
    CALL SET2PE(IPE,0,0,JW,MYSETV)
    ILEN = D%NLEN_M(JW,1,JM)*IFIELD
    IF( ILEN > 0 )THEN
      IOFFS = (D%NSTAGT0B(JW)+D%NOFF_M(JW,1,JM))*IFIELD
      IOFFR = (D%NSTAGT0BW(JW,MYSETW)+D%NOFF_M(JW,1,JM))*IFIELD
      FOUBUF_C(IOFFR+1:IOFFR+ILEN)[IPE]=FOUBUF_IN(IOFFS+1:IOFFS+ILEN)
    ENDIF
    ILENS = D%NLEN_M(JW,2,JM)*IFIELD
    IF( ILENS > 0 )THEN
      IOFFS = (D%NSTAGT0B(JW)+D%NOFF_M(JW,2,JM))*IFIELD
      IOFFR = (D%NSTAGT0BW(JW,MYSETW)+D%NOFF_M(JW,2,JM))*IFIELD
      FOUBUF_C(IOFFR+1:IOFFR+ILENS)[IPE]=FOUBUF_IN(IOFFS+1:IOFFS+ILENS)
    ENDIF
  ENDDO
ENDDO
!$OMP END PARALLEL DO
SYNC IMAGES(D%NMYSETW)
FOUBUF(1:IBLEN)=FOUBUF_C(1:IBLEN)[MYPROC]
```



Trace View - /esfs2/d37/d37/georgem/RAPS12/T159/NOOPARI/u_ifsMASTER_7.otf* - Vampir <@hector-xe6-5>

398.935 s - 399.010 s
75.318 ms

Function Summary

All Processes, Accumulated Exclusive Time per Function Group

Context View

Master Timeline

Property	Value
Display	Master Timeline
Type	Function
Location	Process 0
Function	ltinv ctl$ltinv ctl mod
Function Group	Application
Interval Begin	398.938043 s
Interval End	398.942646 s
Duration	4.6023 ms
Source Loca	ltinv ctl mod F90:3

Process Summary

Similar Processes, Accumulated Exclusive Time per Function

Call Tree

All Processes

Function	Min Inclusive Time	Max Inclusive Time
⊞ 🟩 ltinv ctl$ltinv ctl mod	0.000 s	26.442 ms
⊞ 🟩 ftinv ctl$ftinv ctl mod	0.000 s	52.853 ms
dmapp syncid wait	0.000 s	5.590 µs
⊞ dmapp set rma attrs	0.000 s	5.777 µs
dmapp put	0.000 s	2.995 ms
⊞ dmapp get rma attrs	0.000 s	12.411 µs
dmapp get nb	0.000 s	84.420 µs

Function Legend

- 🟩 Application
- DMAPP
- 🟥 MPI

Schedule for future IFS optimizations in CRESTA

When	Activity
2H2013	**Scaling runs of T3999 model on TITAN (CRESTA INCITE award)** **Initial use of GPUs for IFS (targeting costly LTINV/LTDIR dgemm's)** **Some OpenACC experiments with IFS**
2014	**Further IFS scalability optimizations** • Radiation [wave model, surf scheme] computations in parallel with model • transpose SL data • Use of coarray teams in next Fortran 201X standard • Coarray transfers are still in OMP Critical Sections (do we still need CSs) **Explore use of DAG parallelization (with OMPSs)** • With a toy code representative of IFS **Development & testing of alternative local data structures (minimizing communications) for IFS**

SC12 paper

George Mozdzynski, Mats Hamrud, Nils Wedi, Jens Doleschal, Harvey Richardson, "A PGAS Implementation by Co-design of the ECMWF Integrated Forecasting System (IFS)," High Performance Computing, Networking Storage and Analysis, SC Companion:, pp. 652-661, 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, 2012

Thank you for your attention

QUESTIONS?

Some of the “issues” at the Exascale

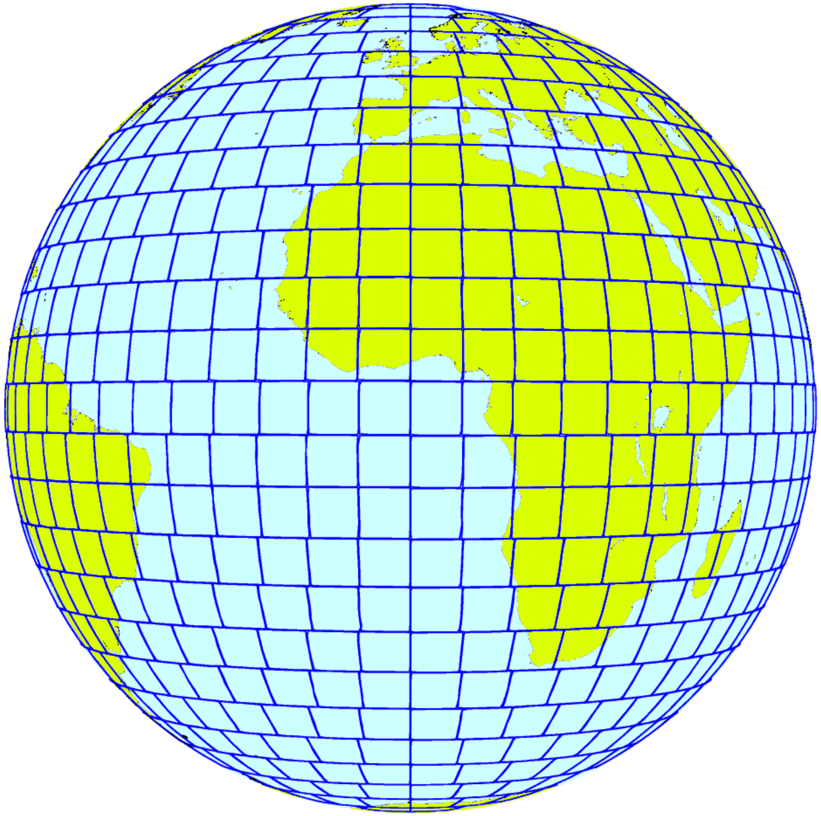
- Power
  - An Exascale computer today would require about a gigawatt ($1B per year)
  - 20 megawatt seen as a limit for governments with deep pockets
  - We expect engineers will solve this problem
- Processors are not getting faster
  - They are getting slower
  - But this is more than compensated by their number (e.g. GPGPUs)
- Reliability
  - Uptime for single system ~ 1 day
  - Implies redundancy of nodes, network, filesystem, no single point of failure
- Scalability of applications
  - Incremental / disruptive solutions / new algorithms / I/O
  - Ensemble methods?

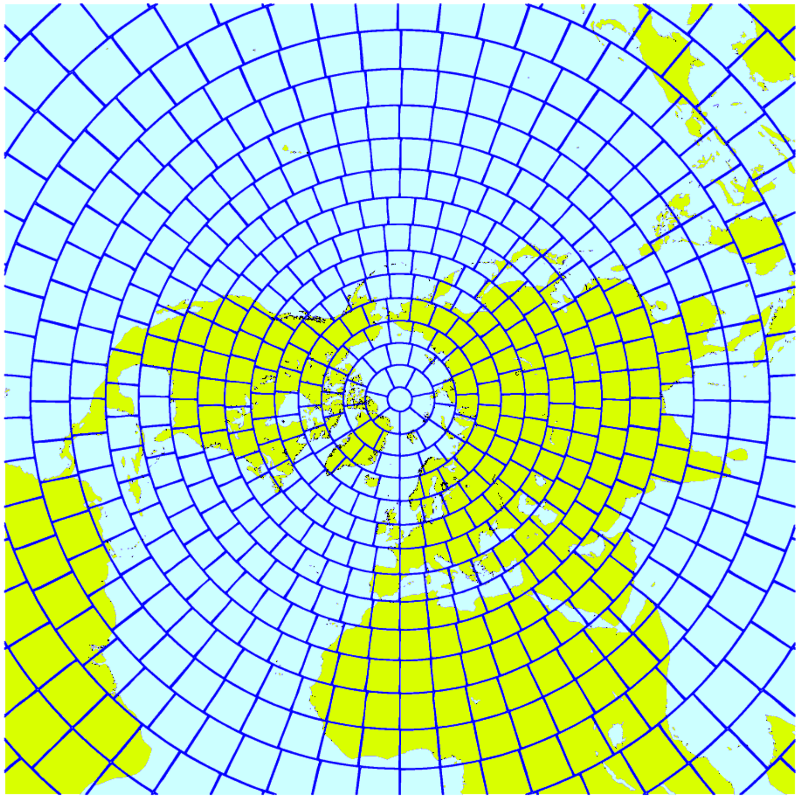
An example of why running a single model at the Exascale will be "challenging"

- Assume the following,
    - model time step of 30 seconds
    - 10 day forecast
    - model on 4M cores
    - max 1 hour wall clock
- 1 step needs to run in under 0.125 seconds
- Using 32 OpenMP threads per task, we will have 128K MPI tasks
- Say we do a simple MPI_SEND from 1 task (e.g. master) to all other 128K tasks
- This will take an estimated 128K x 1 microsec = 0.128 seconds
- Of course we need to use more efficient MPI collectives
- Implies global communications cannot be used, or
- Each task needs to run with 100's or 1000's of threads or GPU cores => max O(10K) MPI tasks, and
- Use of 2D or 3D parallelization

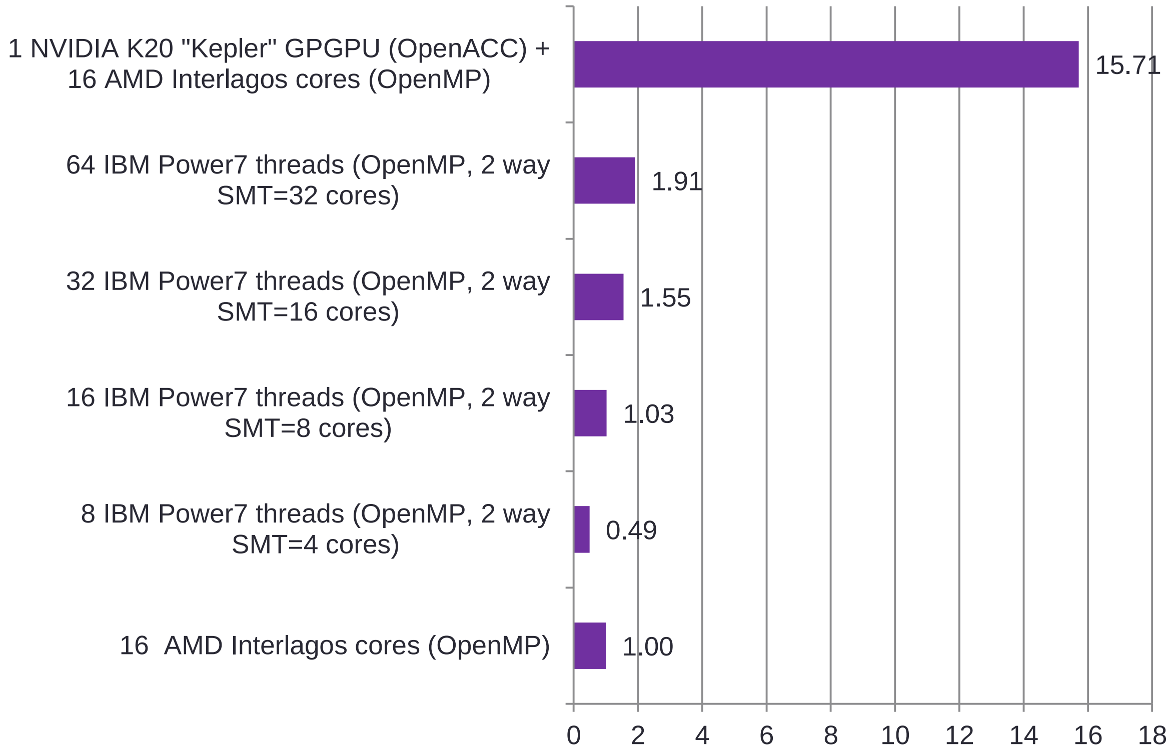
IFS grid point space: “EQ_REGIONS” partitioning **for** 1024 MPI tasks

Each MPI task has an equal number of grid points





Single node performance for md.F90 **
(normalised by wall clock time for 16 AMD Interlagos cores)

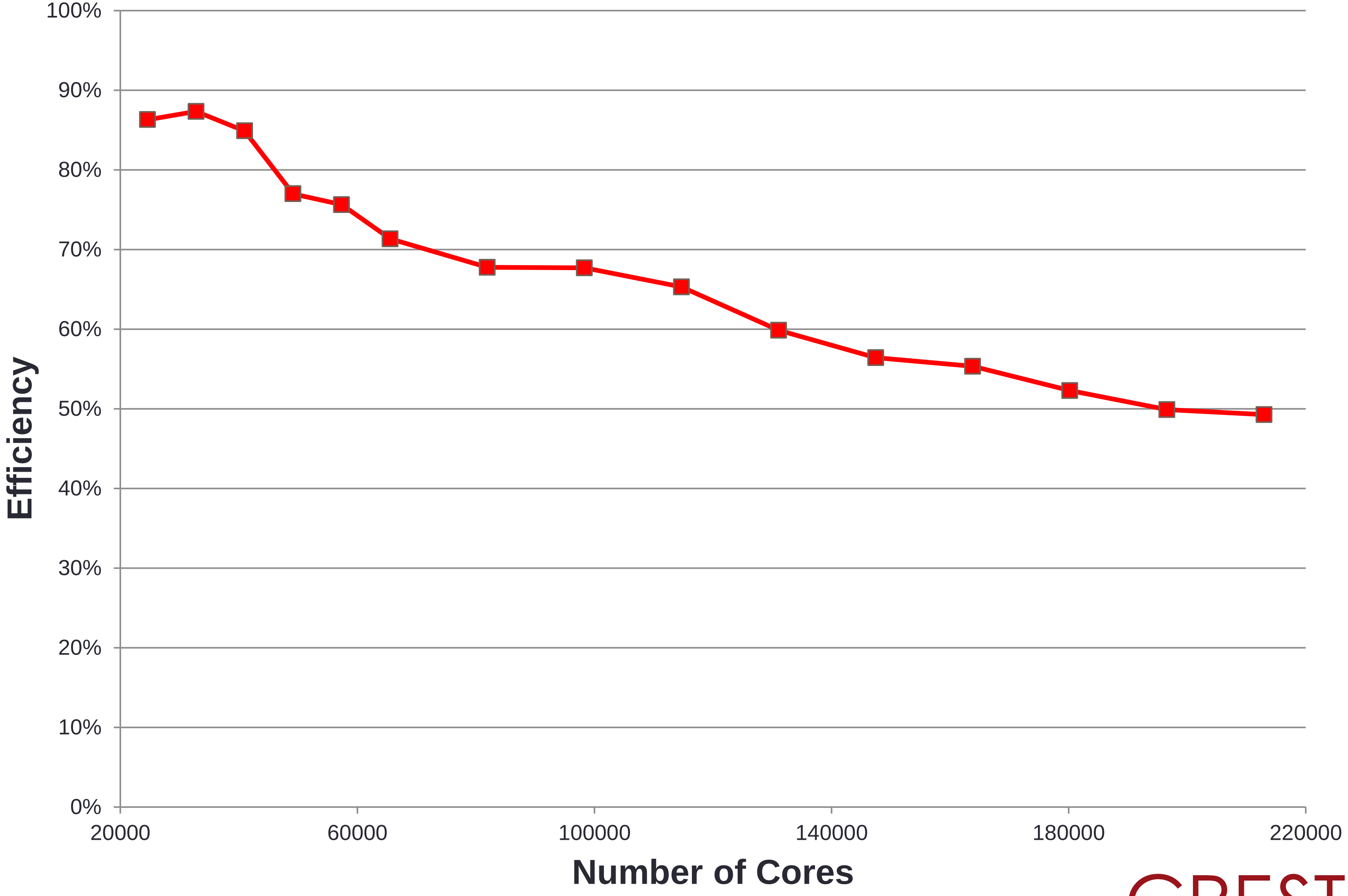


** md.F90 is a small (237 lines) molecular dynamics kernel
Thank you to Alistair Hart (CRAY) for helping me with the OpenACC version
Happy to share OpenMP and OpenACC code, send me an email

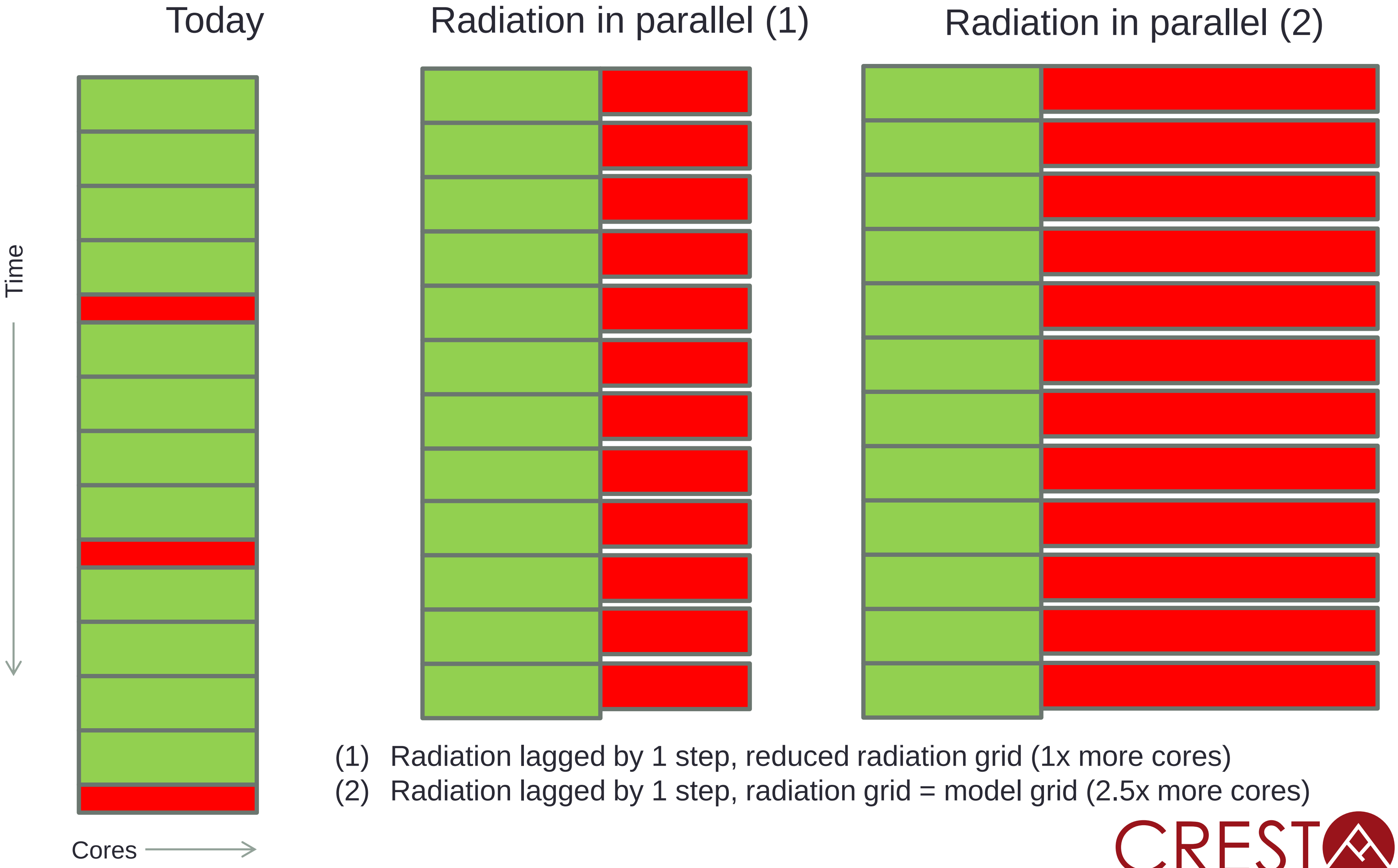
Summary

- Many challenges exist for IFS **applications** to run at the Exascale
- First of these is for hardware vendors to build Exascale computers that are both affordable (cost + power) and reliable
- Ease of programming GPGPU technology will be much easier in the future when there is a single address space for GPGPU cores and conventional cores (if available)
  - Will we need OpenACC in this future?
  - The term GPGPU will disappear in the future
- Our IFS **applications** will require substantial development in the years to come

IFS T3999L137 hydrostatic forecast model efficiency on TITAN RAPS12 IFS (CY38R2), cce=8.1.5, NRADRES=2047, NRADFR=1



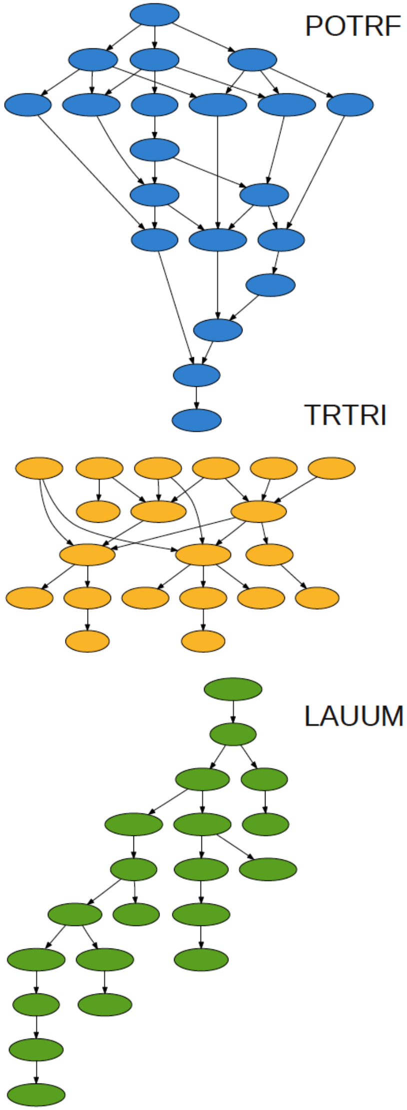
Radiation computations in parallel with model

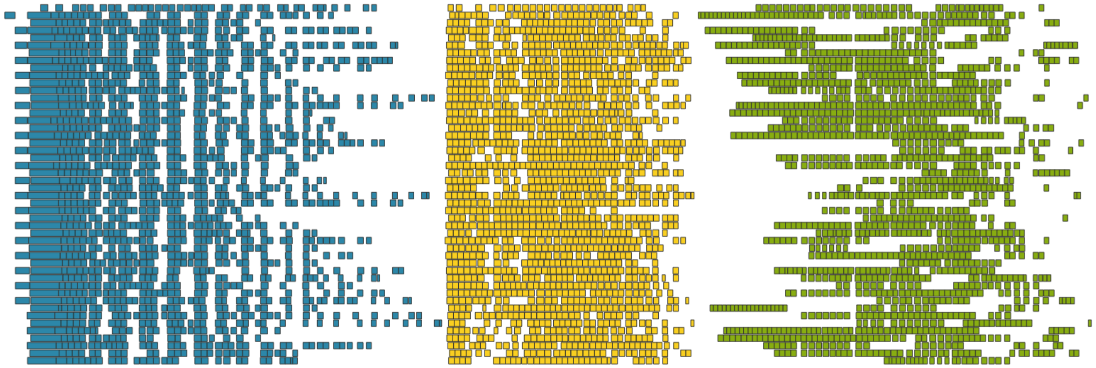


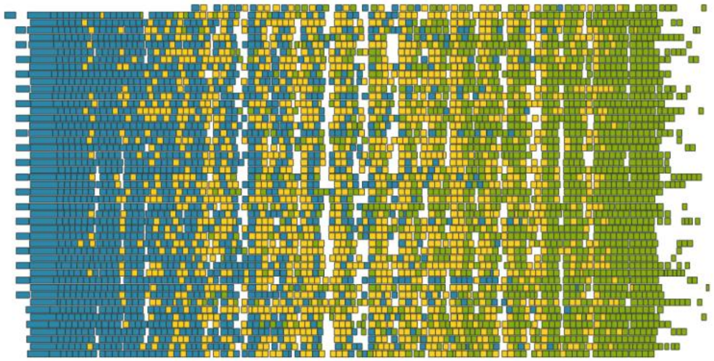
(1) Radiation lagged by 1 step, reduced radiation grid (1x more cores)
(2) Radiation lagged by 1 step, radiation grid = model grid (2.5x more cores)

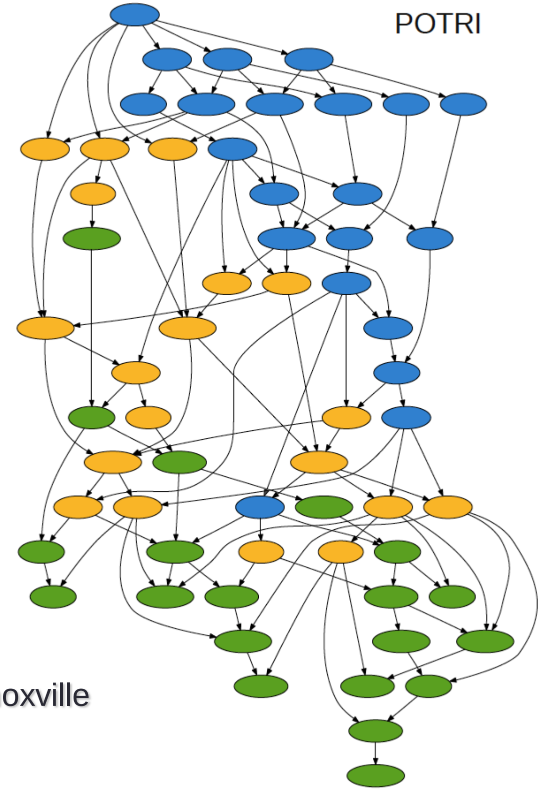
CREST

DAG example: Cholesky Inversion









DAG = Directed Acyclic Graph

Can IFS use this technology?

Source: Stan Tomov, ICL, University of Tennessee, Knoxville

How far can we go with …

technology applied at ECMWF for the last 30 years …

A spectral transform, semi-Lagrangian, semi-implicit (compressible) (non-)hydrostatic model?

-Computational efficiency on and affordability of future HPC architectures ?
-Accuracy and predictability at cloud-resolving scales ?

"The reports of my death have been greatly exaggerated"
Mark Twain

The spectral transform method, dead or alive ?

IFS model coarray developments

Compile with –DCOARRAYS

for compilers that support Fortran2008 coarray syntax

Run with,

```
&NAMPAR1
LCOARRAYS=true,
```

to use coarray optimizations

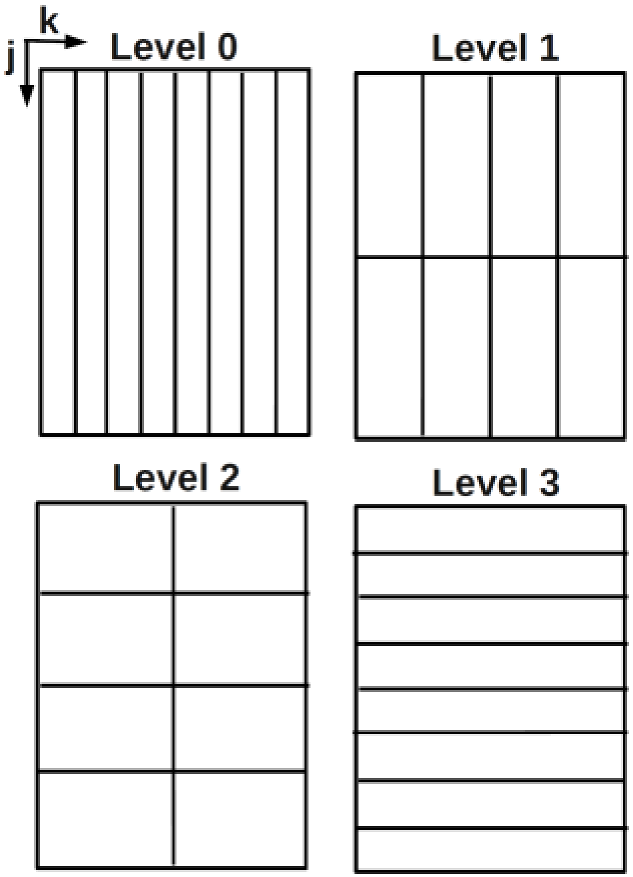
```
&NAMPAR1
LCOARRAYS=false,
```

to use original MPI implementation

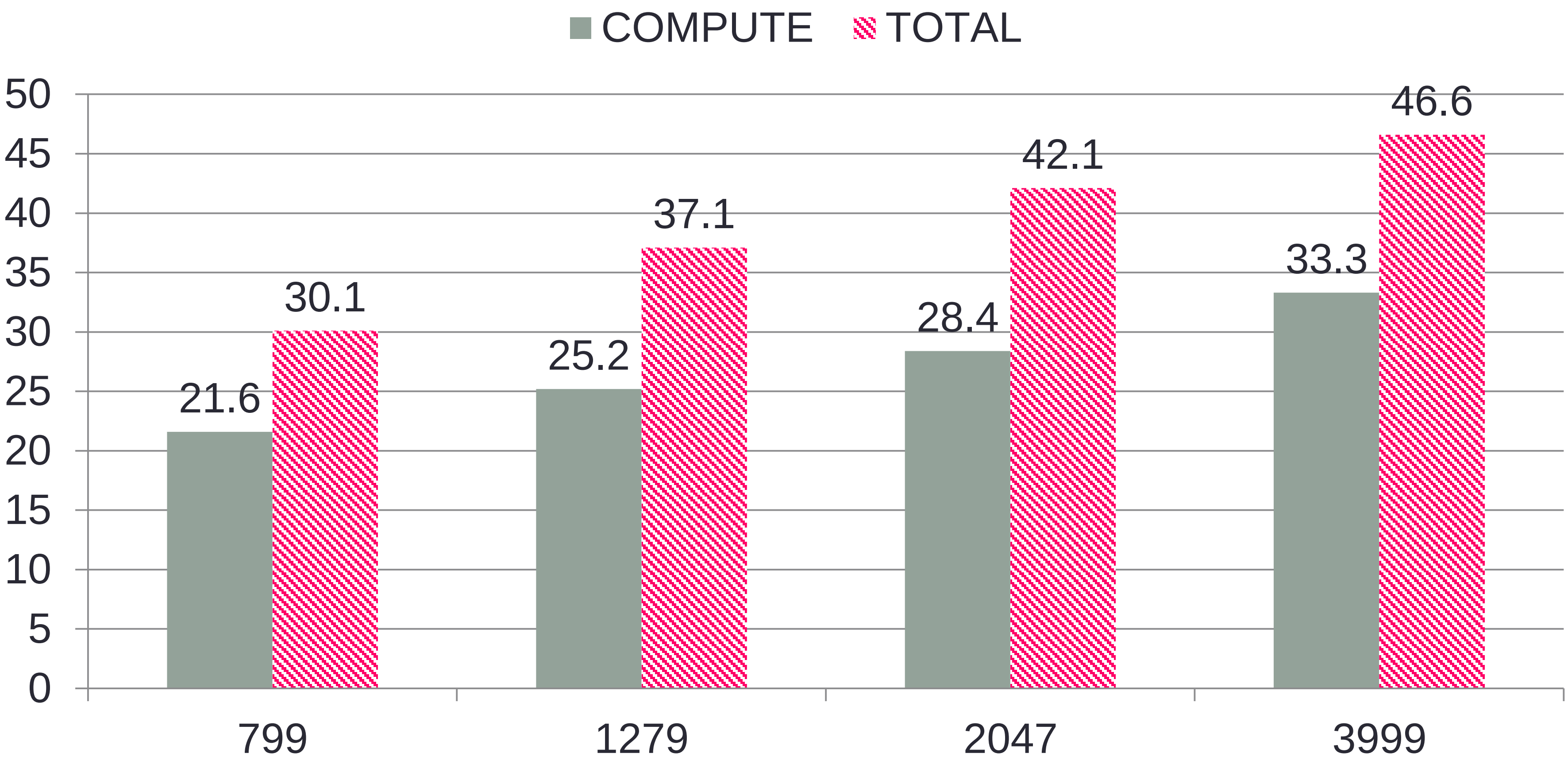
Butterfly algorithm: apply

$$f = S\alpha$$

```
for $l = 0 \rightarrow L$ do
    for all $j, k$ boxes do
        if $l = 0$ then
            $store\ \beta_{0,k} = A_{0,k}\alpha_k$
        else
            $store\ \beta_{l,j,k}$
                $= A_{l,j,k} \times comb\_l\_and\_r\_neighb(\beta, l-1)$
        end if
        if $l = L$ then
            $store\ f_{L,j} = C_{L,j}\beta_{L,j}$
        end if
    end for
end for
```



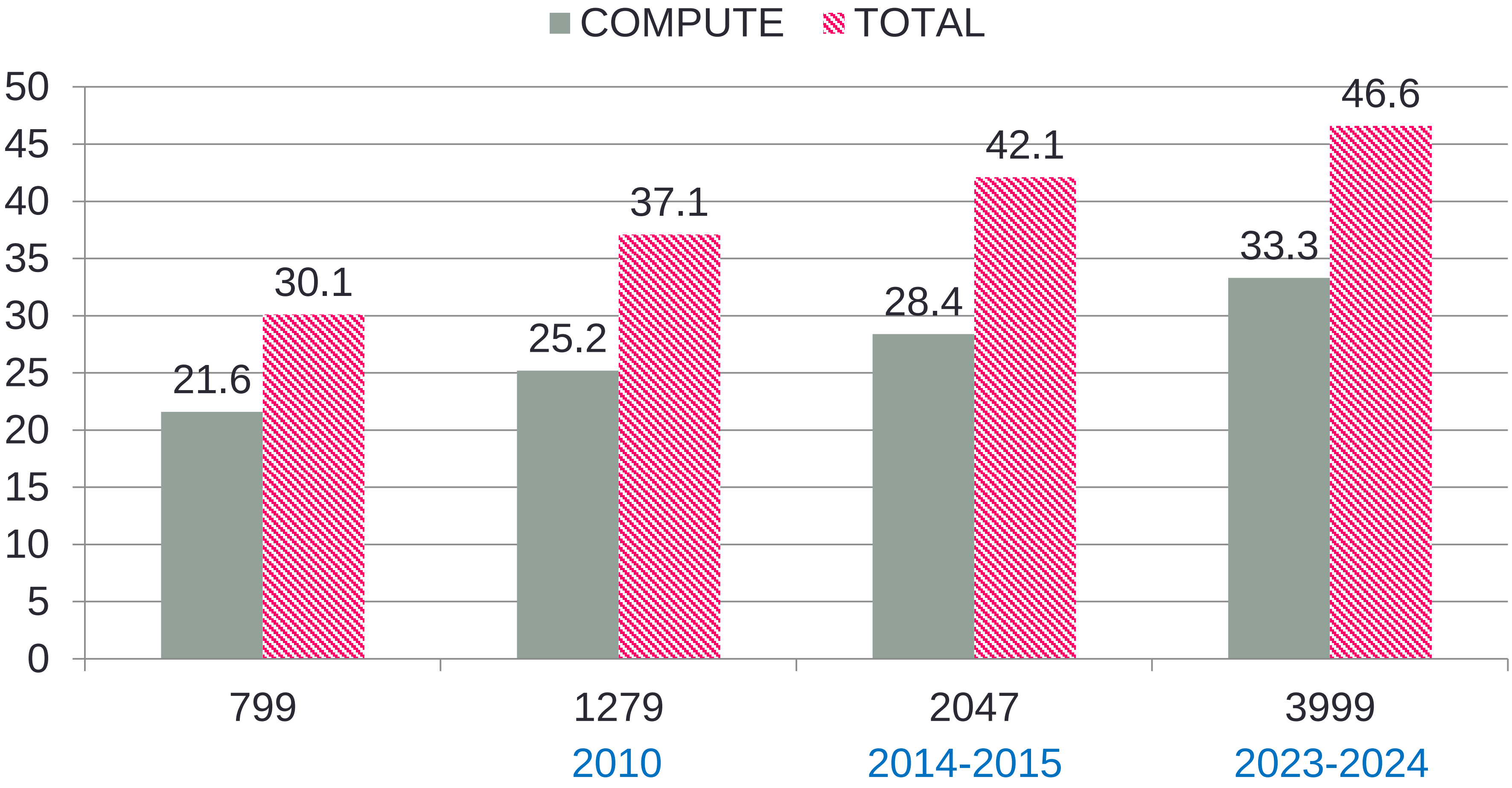
% (of total execution time) cost of spectral part of the model on IBM Power7 (all L91, all NH for comparison); Total includes communications



We expect significant reductions in future cores -> vector instr. / GPU

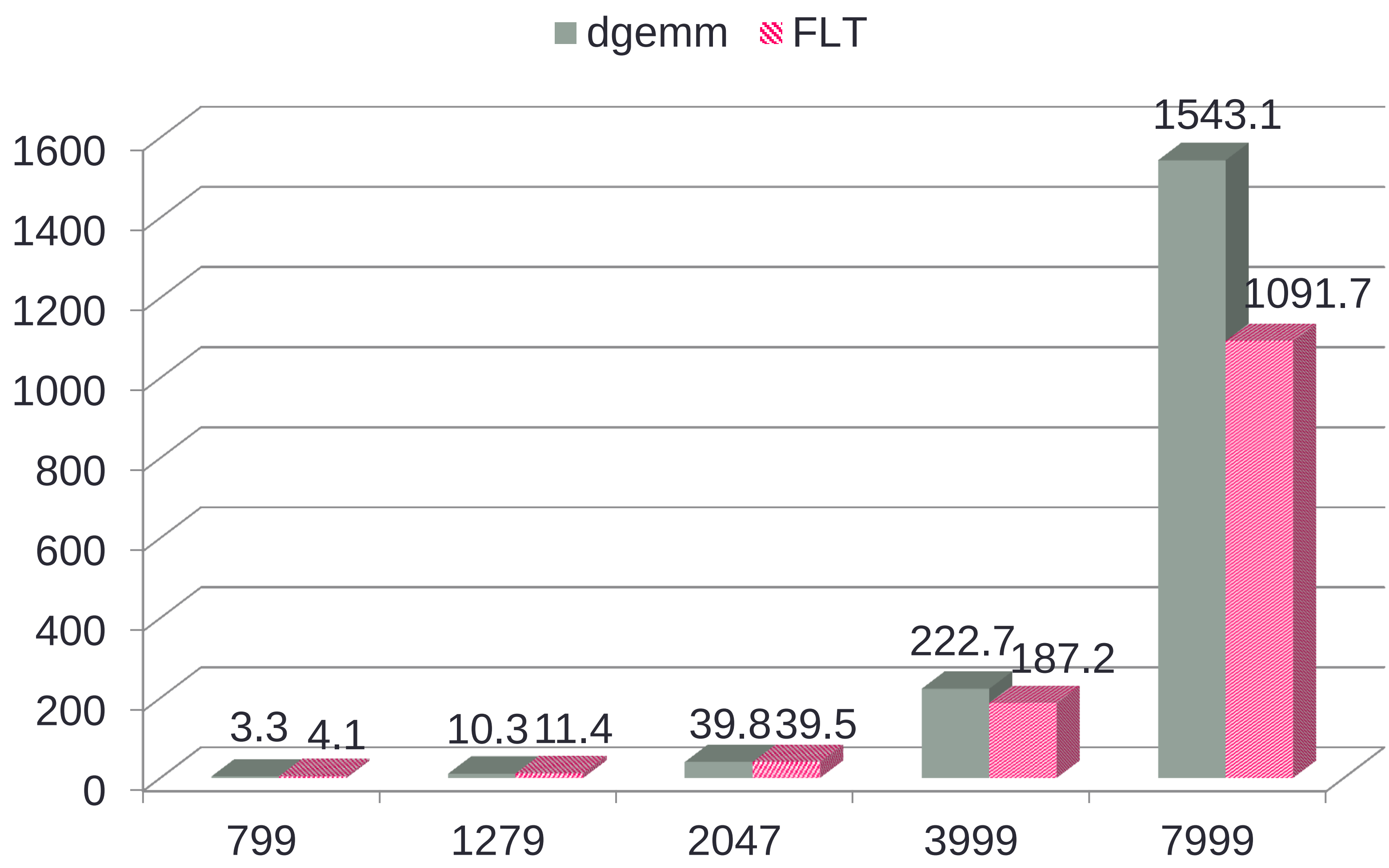
All these can be run with hydrostatic code == ½ of above numbers !

% cost of Spectral Transforms on IBM Power7 (all L91, all NH for comparison)



Expect significant reductions in future cores -> vector instr. / GPU

Average wall-clock time compute cost [milli-seconds] per spectral transform



IFS PGAS Optimisations for Exascale

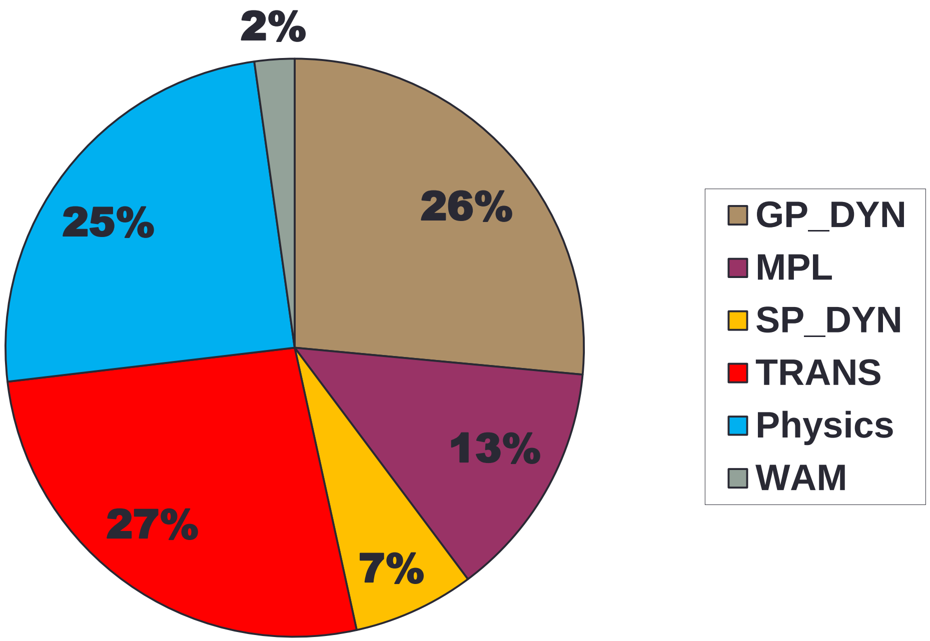
- IFS PGAS optimisations in the CRESTA project
  - Involve use of Fortran2008 coarrays (CAF)
  - Used within context of OpenMP parallel regions
- Overlap Legendre transforms with associated transpositions
- Overlap Fourier transforms with associated transpositions
- Rework semi-Lagrangian communications
  - To substantially reduce communicated halo data
  - To overlap halo communications with SL interpolations
- Explore GPU and Vector technology for further computational speed-ups of matrix-matrix multiplies

Numerical solution

- Two-time-level, semi-implicit, semi-Lagrangian.
- Semi-implicit procedure with two reference states, with respect to gravity and acoustic waves, respectively.
- The resulting Helmholtz equation can be solved (subject to some constraints on the vertical discretization) with a *direct spectral method, that is, a mathematical separation of the horizontal and vertical part of the linear problem in spectral space, with the remainder representing at most a pentadiagonal problem of dimension NLEV^2. Non-linear residuals are treated explicitly (or iteratively implicitly)!*

(Robert, 1972; Bénard et al 2004,2005,2010)

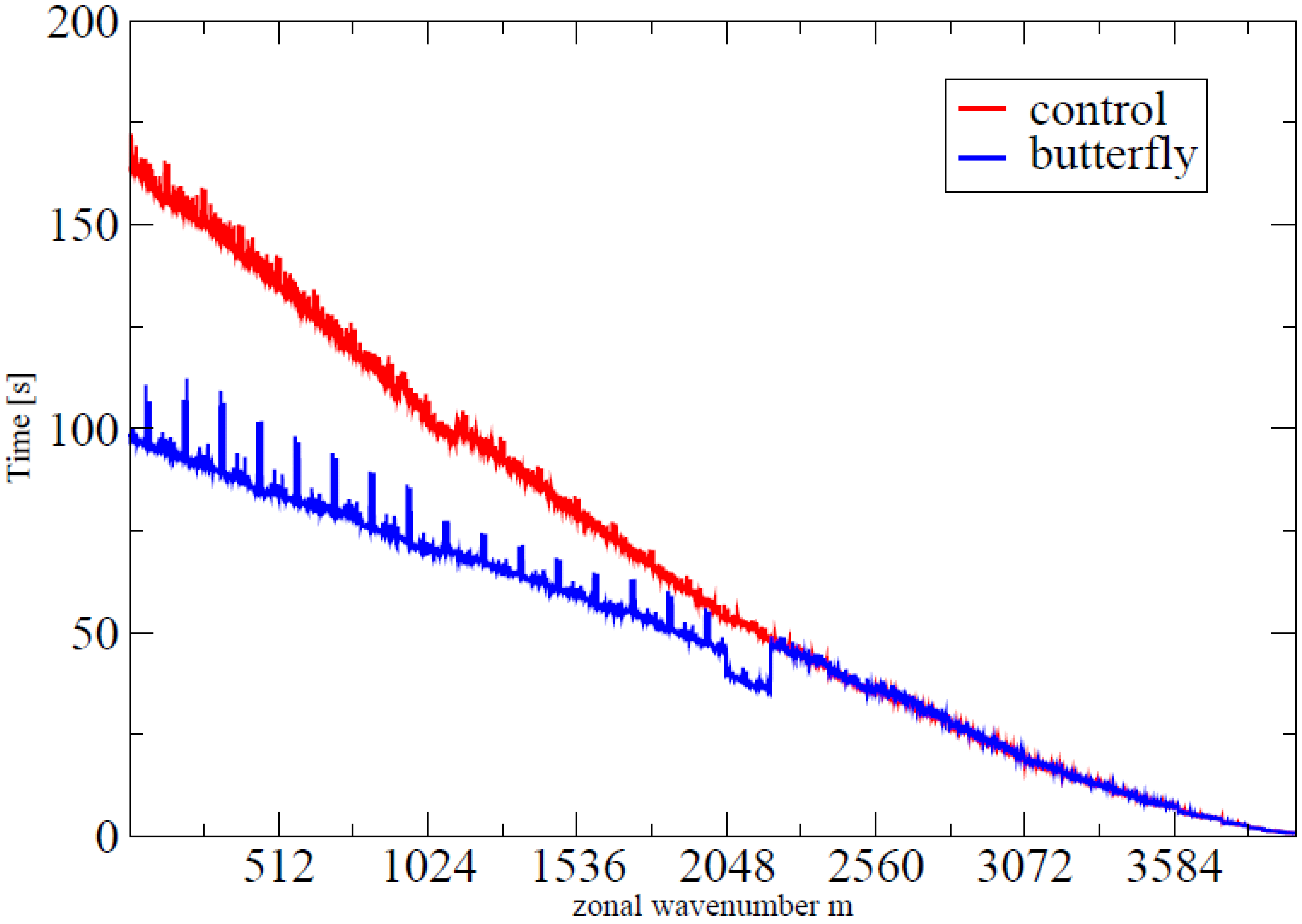
NH IFS T_L3999 L91 (5 km) on IBM Power7 with FLT



SP_DYN was 23 percent for this model configuration, and is now 7 percent. Improvement due to exposing 'greater OpenMP parallelism' from 4K threads to a maximum of 4K * 91 threads ; in this case 16K threads.

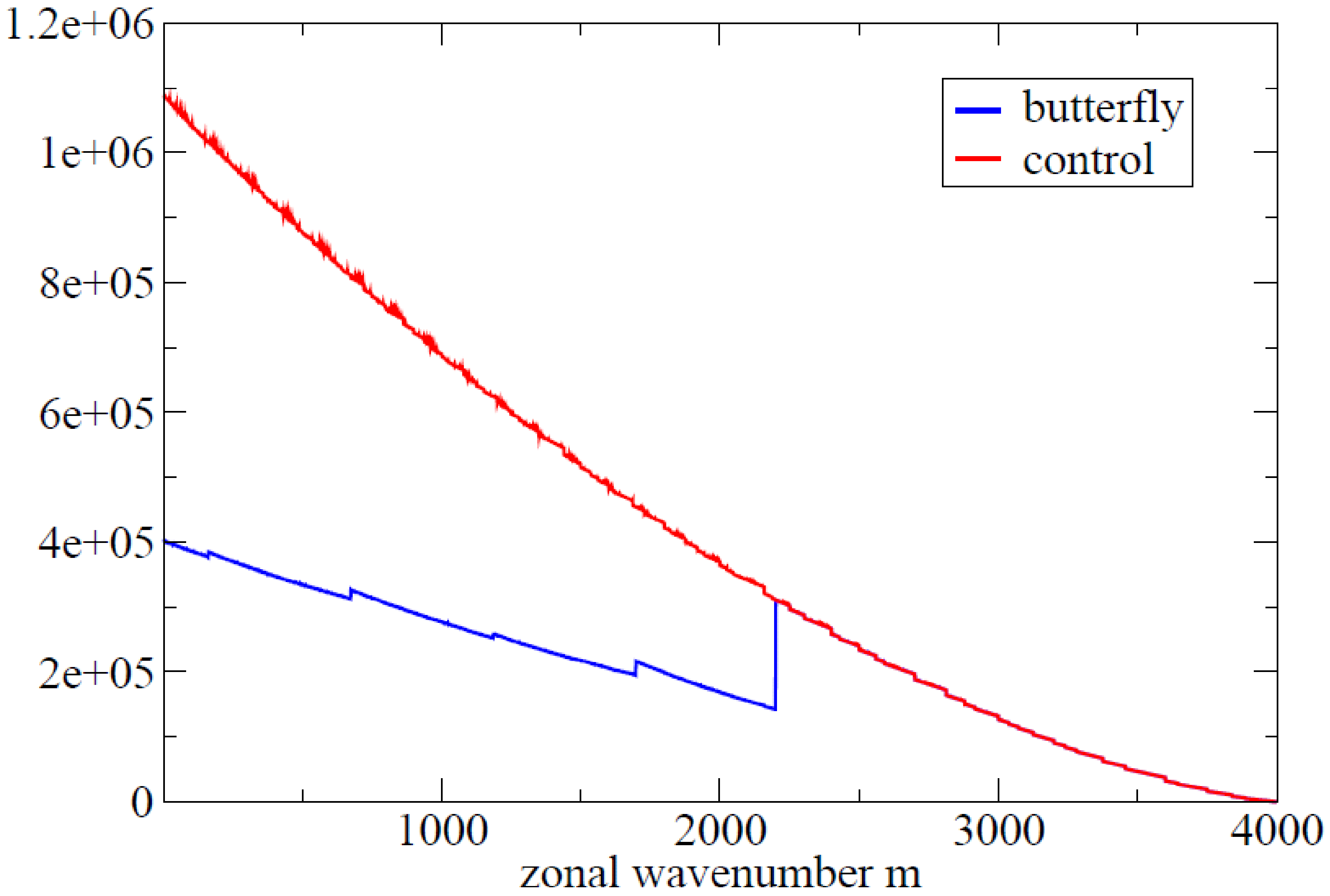
TSTEP=180s, 3.1s/iteration
Using 1024 tasks x16 OpenMP threads
10 day forecast ~ 4 hours for this config

T3999 6h forecast - inverse transforms: CPU time vs. wave



CREST

T3999 6h forecast - inverse transforms: Floating point operations vs. wave number



Exascale problem projections

- To run a T7999 L137 forecast (~2.5km) may require approximately 1-4 million processors (of current technology) to run in one hour
- At the same time 1-4 Million processors could run a 50 member ensemble of T3999 L137 in the same hour
- But first we have to be able to run a T3999 L137 forecast efficiently in one hour!