Software co-design for the exascale

Erwin Laure

Director PDC
KTH, Stockholm, Sweden

Based on Material from Mark Parsons (EPCC)
and George Mozdzynski (ECMWF)

CRESTA

- **C**ollaborative **R**esearch into **E**xascale **S**ystemware, **T**ools and **A**pplications
- Developing techniques and solutions which address the most difficult challenges that computing at the exascale can provide
- Focus is predominately on software not hardware
- Funded via FP7 by DG-INFSO
- Project started 1st October 2011
- Three year duration
- 13 partners, EPCC project coordinator
- €12 million costs, €8.57 million funding

CRESTA

Partnership

- Consortium
  - Leading E owners and
    - EPCC –
    - HLRS – rsity – Abo,
    - CSC – E
    - KTH – S – Jyvaskyla,
  - A world lea ondon –
    - Cray UK
  - World lead UK
    - Technisc – Paris, France (Vampir)
    - Allinea L rmany
      weden
      ustutt – Stuttgart, Germany

CRESTA

Getting to the exascale

- The road to exascale is a complex one
- We can't do it simply by building hardware
- Many vendors understand this – in the USA they've stopped talking about exascale and are focussing on "*extreme computing*"
- The technologies we need are becoming clearer – a number of which are part of DEEP and MONT BLANC
- However, we are still not taking the amount of parallelism at the Exascale seriously

Building a 20MW Exascale computer

- Based on current technology roadmaps Exascale systems will be impossible to build below 50MW
  - GPUs and Xeon Phi plus traditional multi-core microprocessors, memory hierarchies and even with embedded interconnects cannot get to 20MW
- The Exascale exposes the yawning gap between modern data flow hierarchies inside HPC systems and Amdahl's "laws" of a well balanced computer
- The solution will be to rebalance systems – smaller, simpler processors with faster, simpler memory hierarchies and communications links – all energy efficient
- But these solutions **INCREASE** the amount of parallelism
  - Today's best scales to 92 million cores and 525MW at the Exascale
- Slower better balanced cores means parallelism at the 500 million – 1 billion thread scale

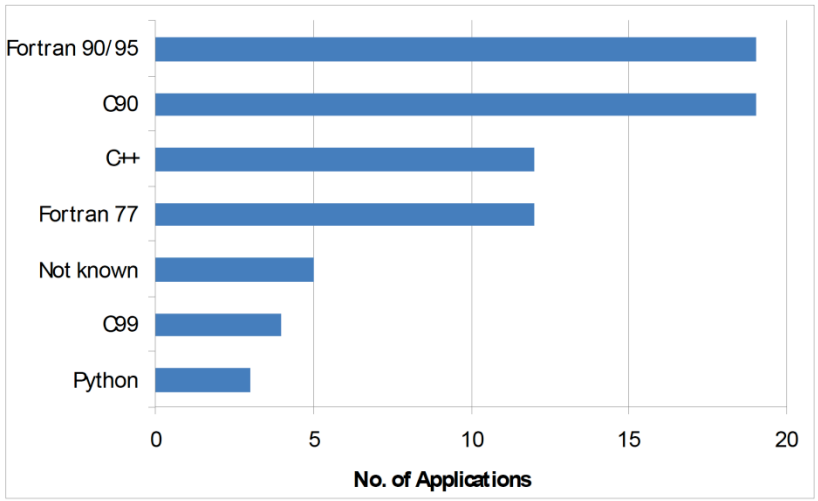
CRESTA

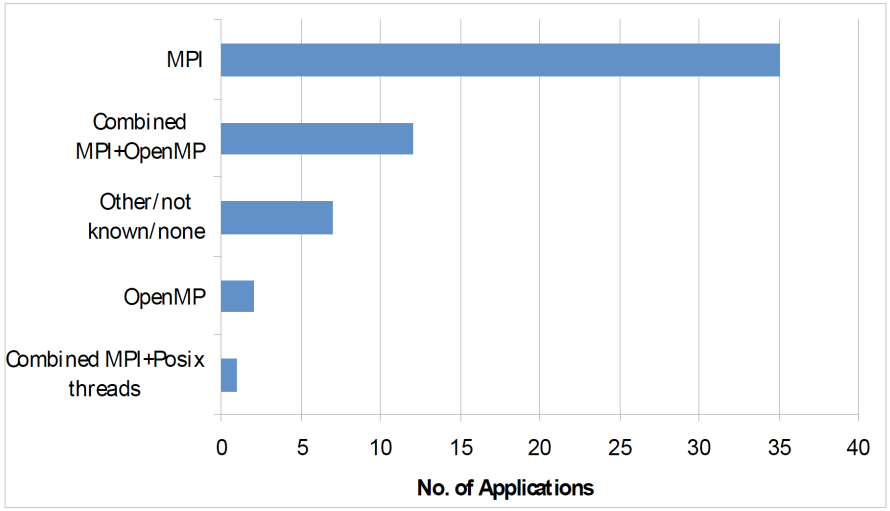
The Exascale software challenge

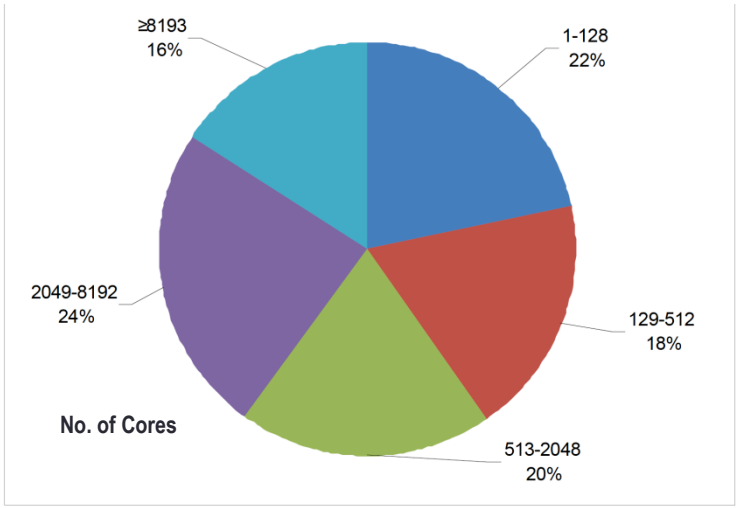
- Few mathematical algorithms are designed with parallelism in mind
  - … parallelism is "just a matter of implementation" is the mind-set
- Scale of parallelism has been remarkably stable for 20 years
  - 1994 Cray T3D 512 cores v. 2010 IBM P-Series 2,580 cores
- Most codes have incrementally approached each new technology – but this has mean much duplication of effort as codes are individually optimised
- Exascale levels of parallelism are a different domain
  - Without fundamental algorithmic changes, progress in many areas will be limited … and not justify the investment in Exascale systems
- We need
  - Research into new mathematical models and parallelism
  - A properly funded European software for modelling and simulation programme
  - Better tools and training – more researchers and practitioners
  - A focus on good software engineering for parallelism
- If we get this right the benefits will extend across all computing – parallelism is pervasive – from your Audi to your iPhone to your Cray

Application scalability

- We have a programmability problem *today* at the Petascale with application scalability …







Figures from a study of the 57 leading applications used in Europe by the PRACE Project

CRESTA

Meeting the software challenge

- Ensemble computing has its place at the exascale – look at molecular dynamics or numerical weather forecasting for proof
- But for some applications we need to consider how to make them scale to 100 million+ parallel threads
- For these applications we need to rethink how we model and simulate their physical problems
- In many cases we need to rethink how we express differential calculus on these computing devices
- Far too much exascale research is focussing on incremental improvements to algorithms and parallelisation technologies
- We will not succeed at the exascale unless we rethink our approach
- The hardware will almost certainly appear … will we have codes to run on it?

CRESTA

Key principles behind CRESTA

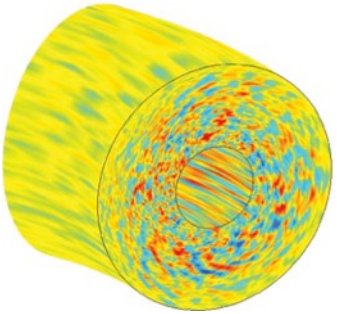
- Two strand project
  - Building and exploring appropriate *sy*
  - Enabling a set of key *co-design* appli
- Co-design is at the heart of the proje
  - provide guidance and feedback to the
  - integrate and benefit from this development cyclical process
- Employing both incremental and disruptive solutions
  - Exascale requires both roaches
  - Particularly true for appl limit of scaling today
  - Solutions will
- Committed to c

Disruptive approach
- Work with co-design applications to consider alternative algorithms
- Crucial we understand maximum performance before very major application redesigns undertaken

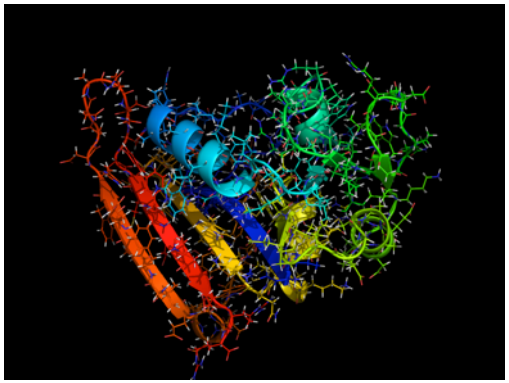
Incremental approach
- Through optimisations, performance modelling and co-design application feedback
- Look to achieve maximum performance at Exascale and understand limitations e.g. through sub-domains, overlap of compute and comms

CRESTA

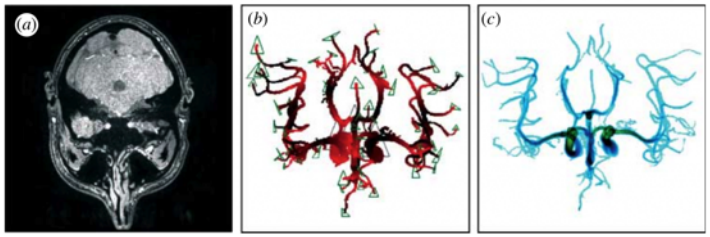
Getting to the exascale



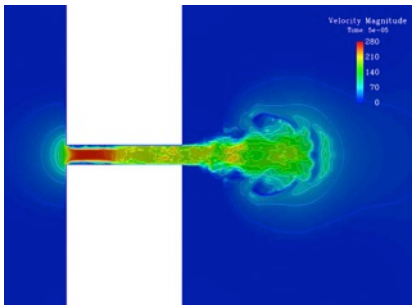
Elmfire: exploring radical memory refactoring, hybridisation



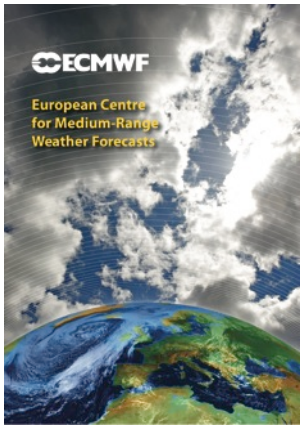
GROMACS: exploring new modelling methods to couple strong scaling on realistic problem sizes with ensemble computing



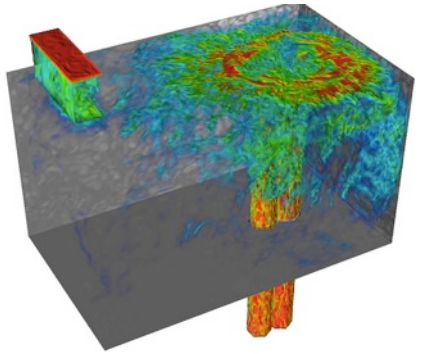
HemeLB: exploring hybridisation, pre- and post-processing and load balancing



OpenFOAM: exploring new solvers and IO optimisation



IFS: exploring new communication models, task-graph parallelism and acceleration



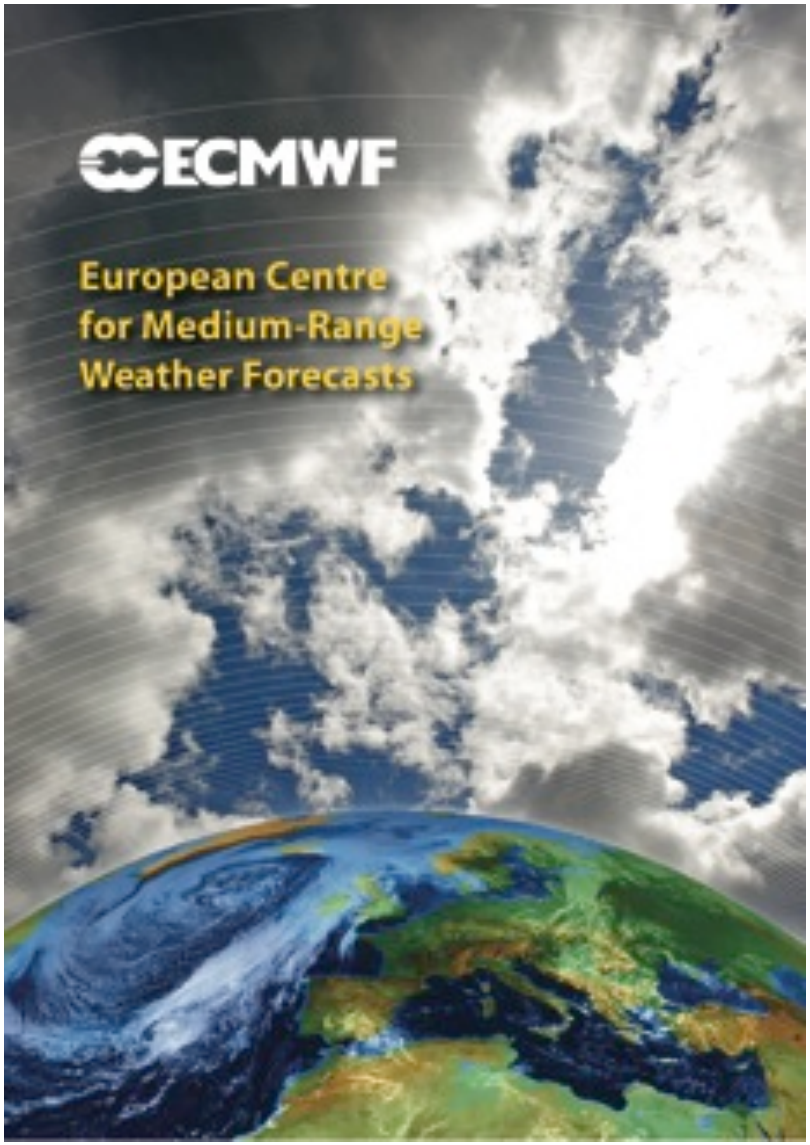
NEK5000: exploring adaptive mesh refinement and acceleration

CRESTA

Can these applications get to the Exascale?

- These applications and tools represent the real-word of HPC
- All do a fraction of a Petaflop today (some better than others)
- All are working in CRESTA to understand how they get to an Exaflop
  - CRESTA is unusual in that it is funding them to explore these thoughts
  - People assume application owners have core funding for this – most don't – much application development is "hidden" in science projects
- Already seen direct decisions being taken
  - Some CRESTA applications need rewritten to get the Exascale
  - Others can see how they will weak-scale – and are looking at what that means for their science
- Applications have a long lifetime – 30+ years – so rewriting is a very costly, major decision

CRESTA and IFS

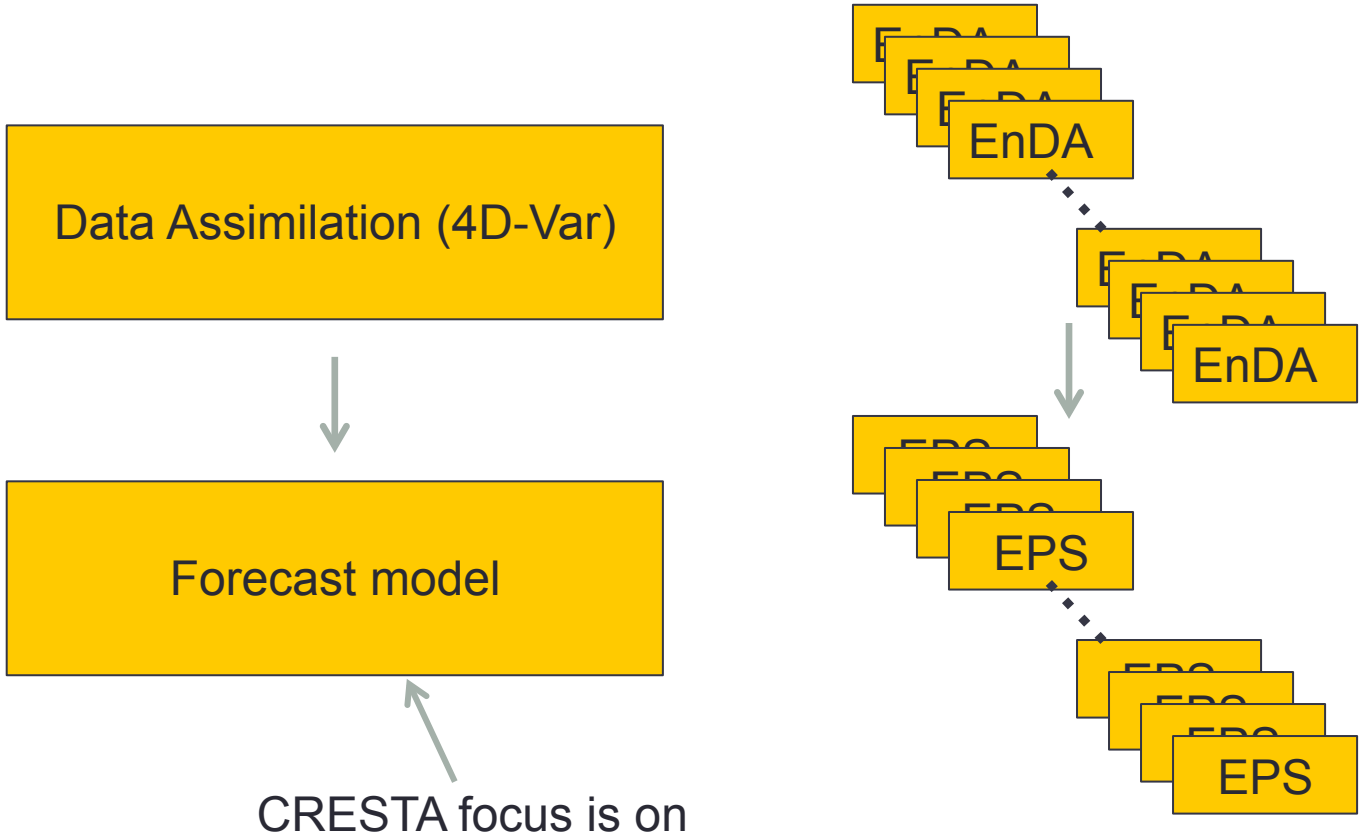


CRESTA

IFS : Key operational workload

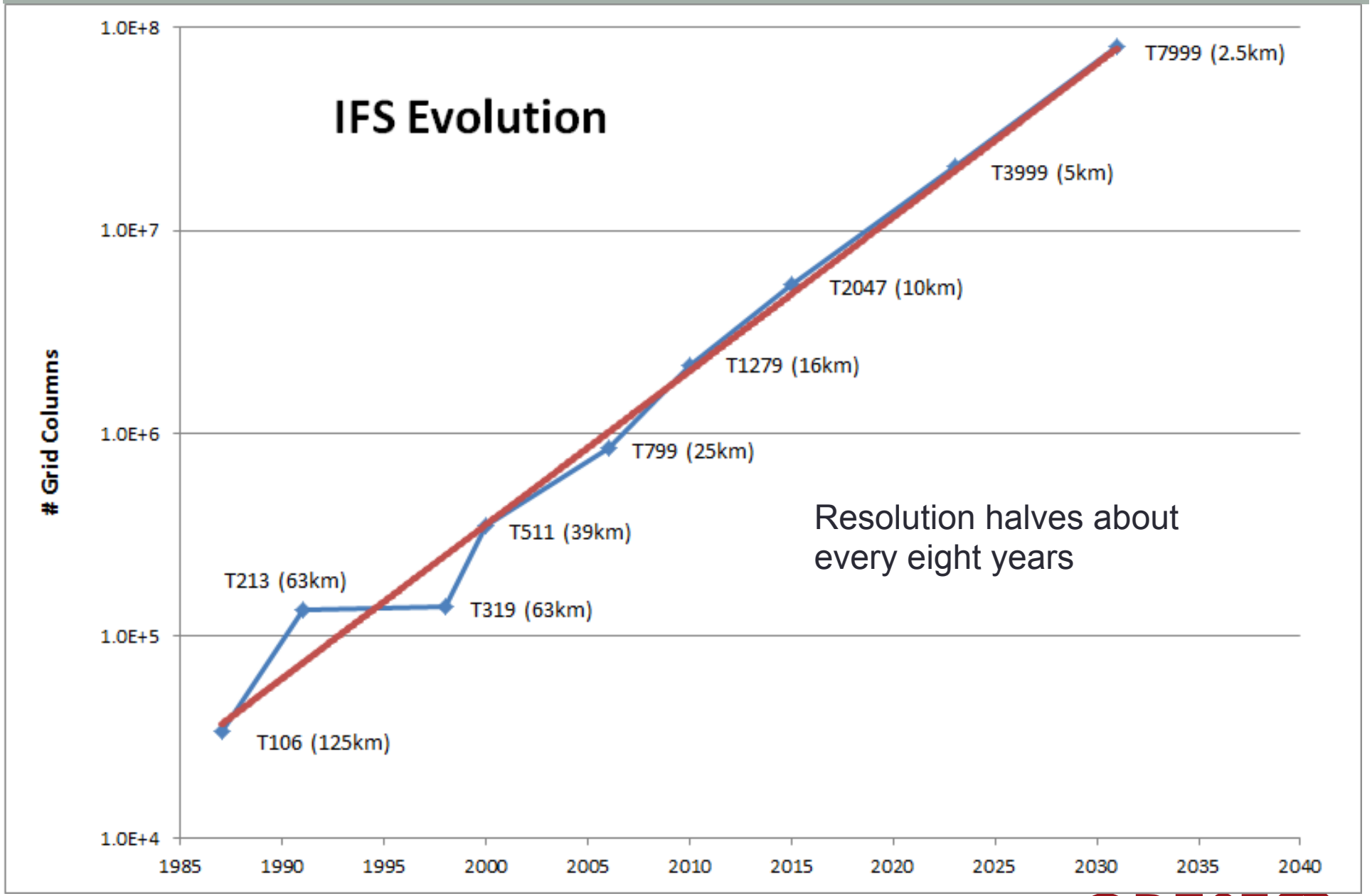
Higher Resolution
(single jobs)

Lower Resolution
(ensembles)

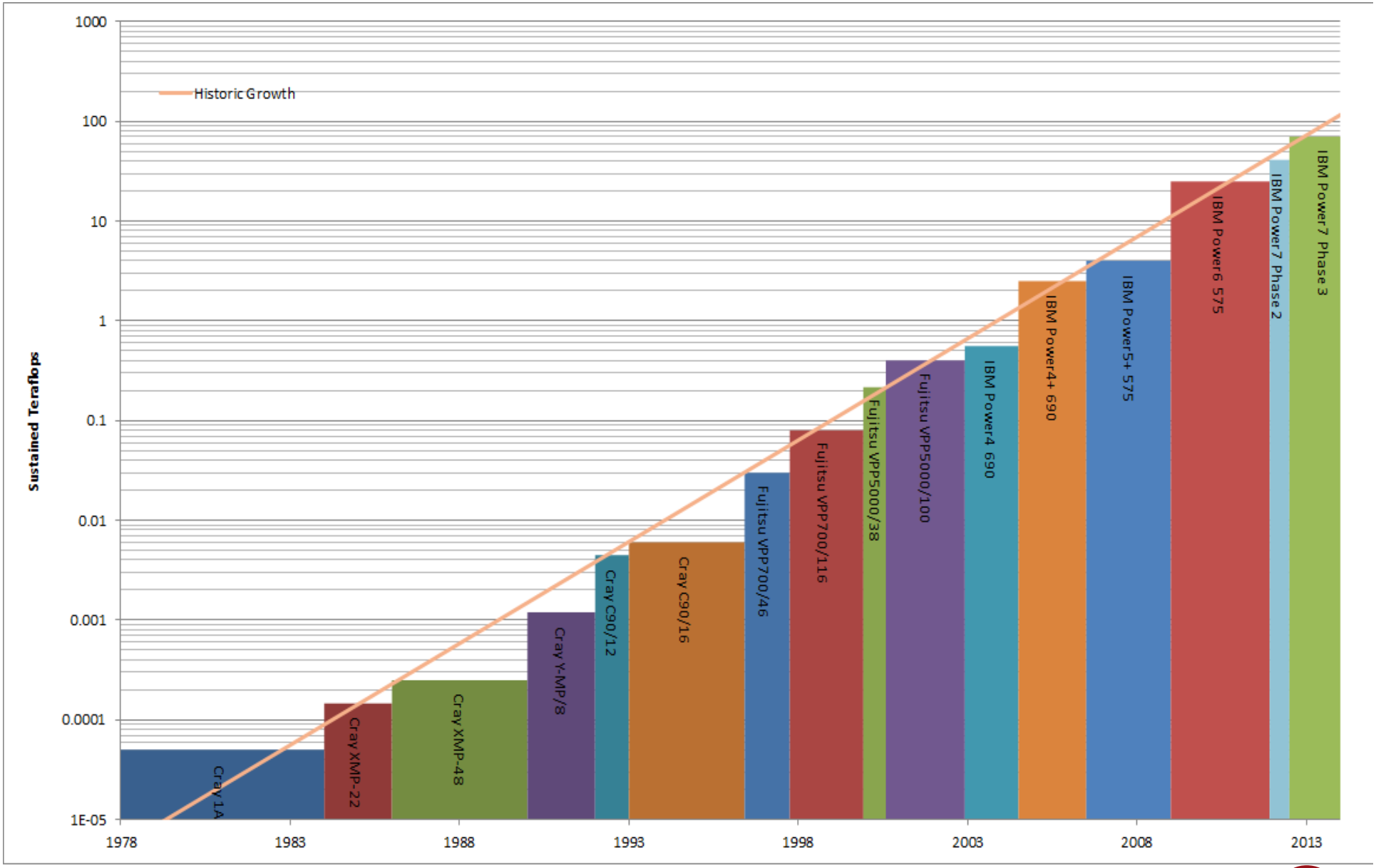


CRESTA focus is on
Forecast Model

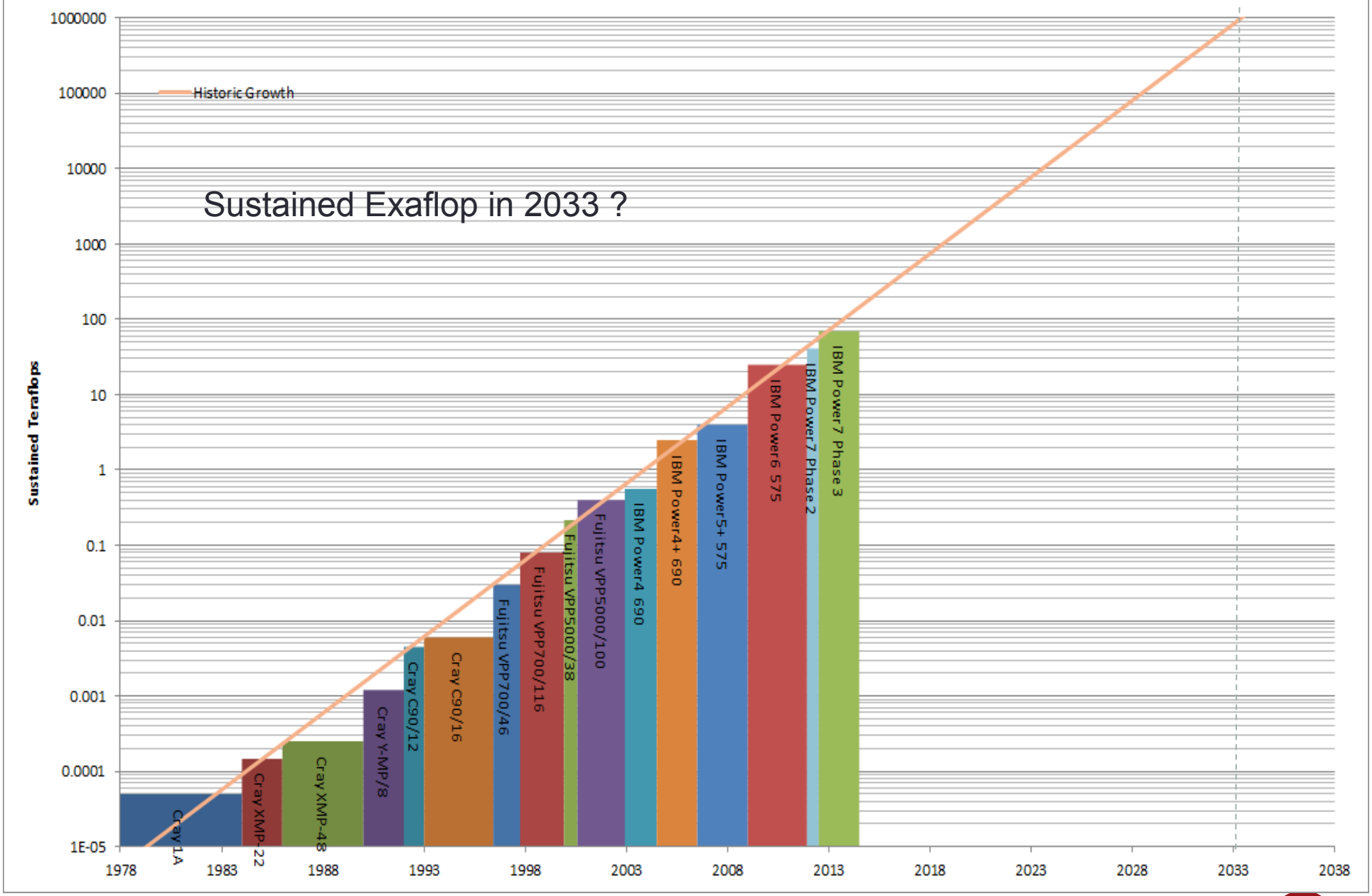
CRESTA


IFS Evolution
Grid Columns
1.0E+8
1.0E+7
1.0E+6
1.0E+5
1.0E+4
1985
1990
1995
2000
2005
2010
2015
2020
2025
2030
2035
2040
T7999 (2.5km)
T3999 (5km)
T2047 (10km)
T1279 (16km)
T799 (25km)
T511 (39km)
T319 (63km)
T213 (63km)
T106 (125km)
Resolution halves about every eight years

CREST


Sustained Teraflops
1000
100
10
1
0.1
0.01
0.001
0.0001
1E-05
1978
1983
1988
1993
1998
2003
2008
2013
Historic Growth
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

CREST


Sustained Exaflop in 2033 ?
Historic Growth
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

CREST

"At the start of the CRESTA project the maximum number of cores that an IFS model had run on was less than 10,000"

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

CRESTA

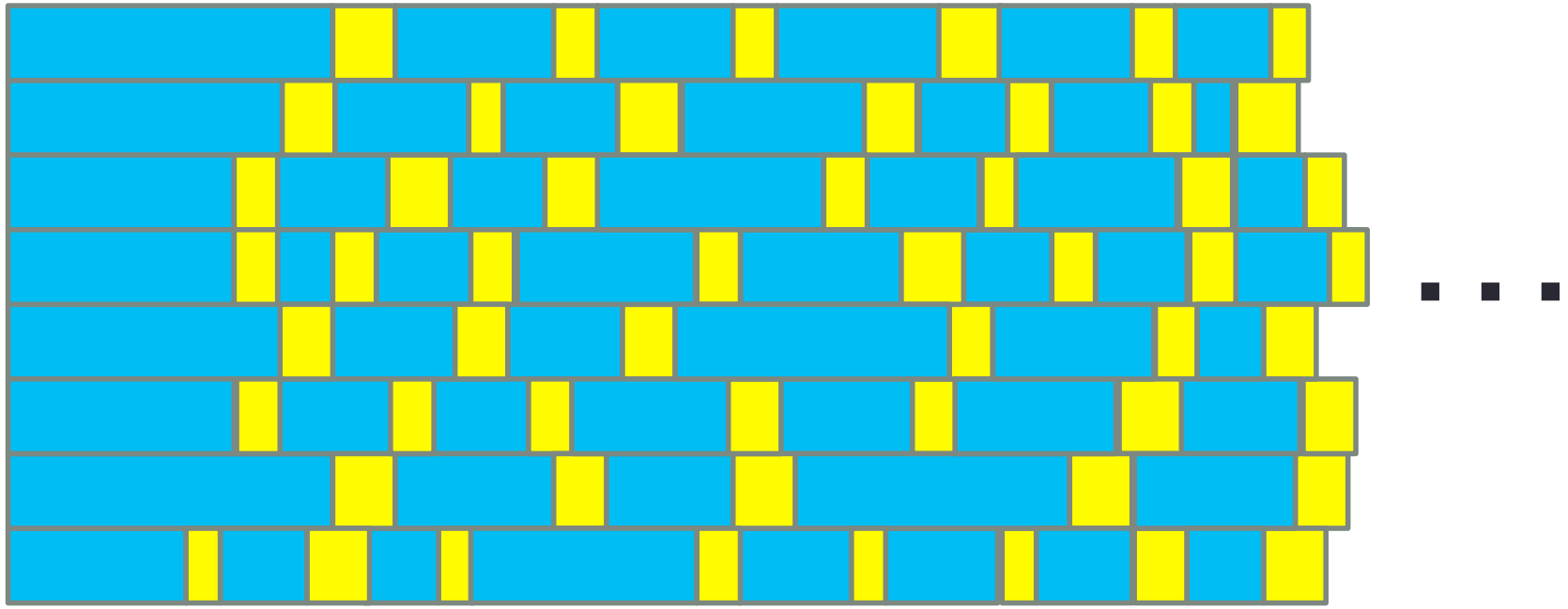
IFS Optimisations for ExaScale & Co-design

- All IFS 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
- CAF co-design team
  - caf-co-design@cresta-project.eu
  - ECMWF – optimise 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
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

CRESTA

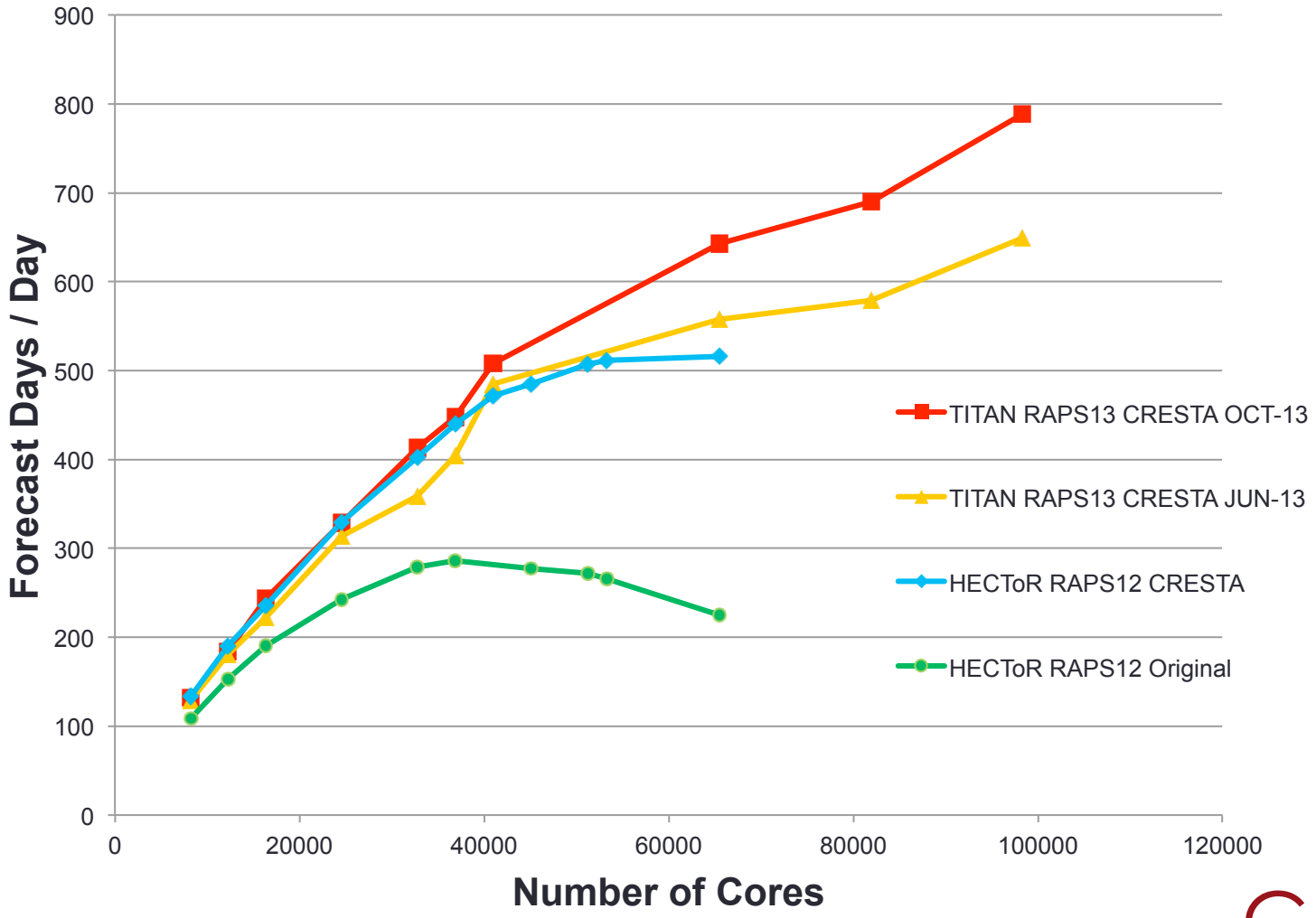
Overlap Legendre transforms with associated transpositions/3 (LTINV + coarray puts)



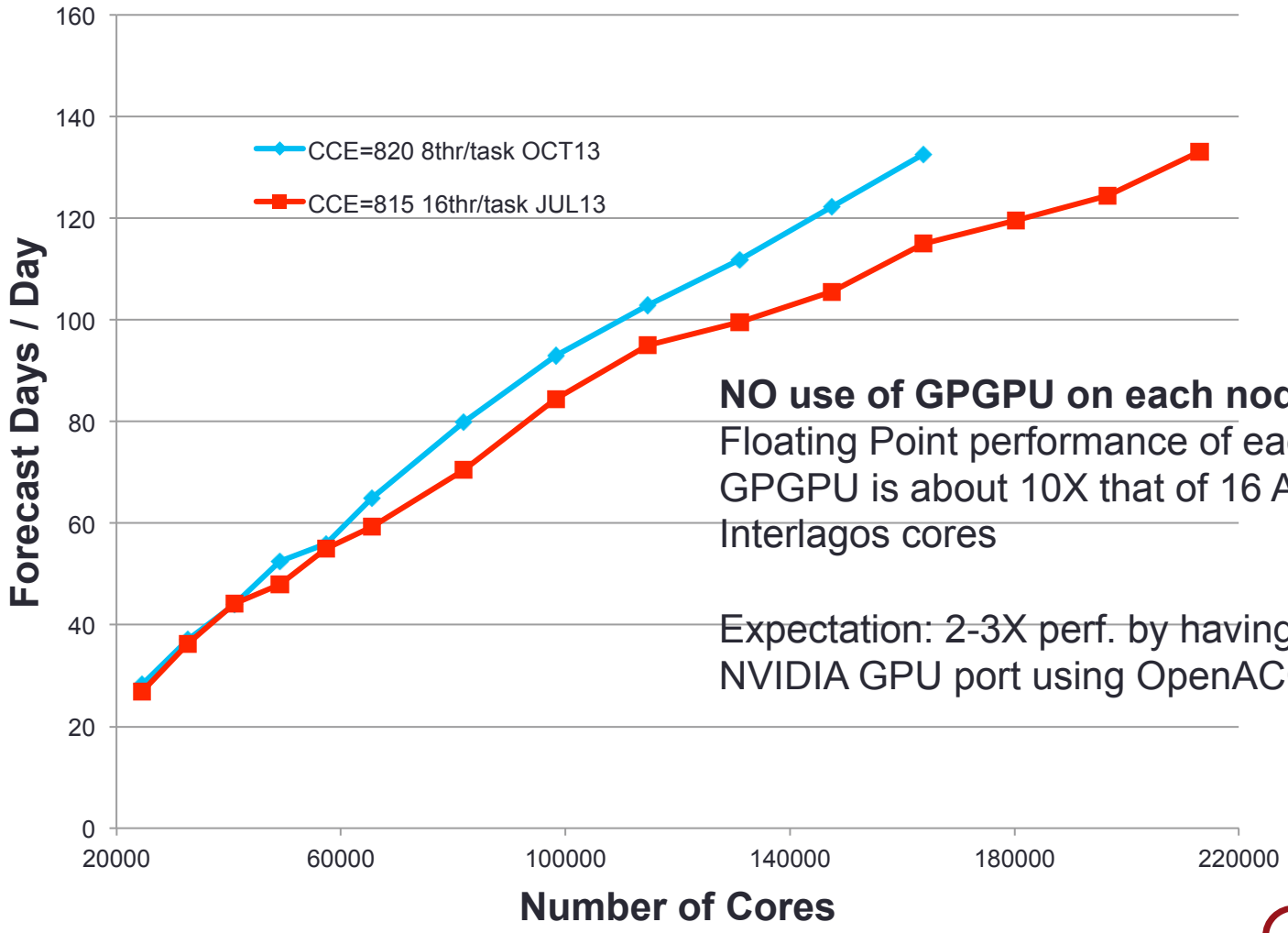
Expectation is that compute (LTINV-blue) and communication (coarray puts-yellow) overlap in time. We can now see this with an extension to vampir developed in CRESTA

CRESTA

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



IFS T3999L137 hydrostatic forecast model performance on TITAN
RAPS13 IFS (CY38R2), 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

Expectation: 2-3X perf. by having a NVIDIA GPU port using OpenACC

CRESTA

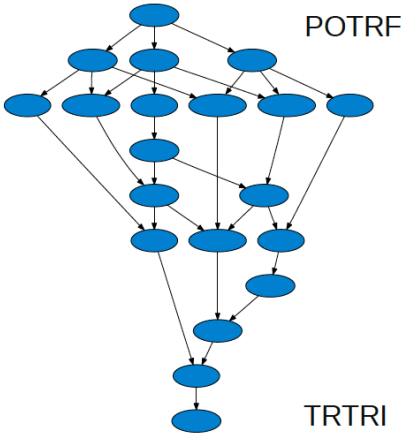
Summary

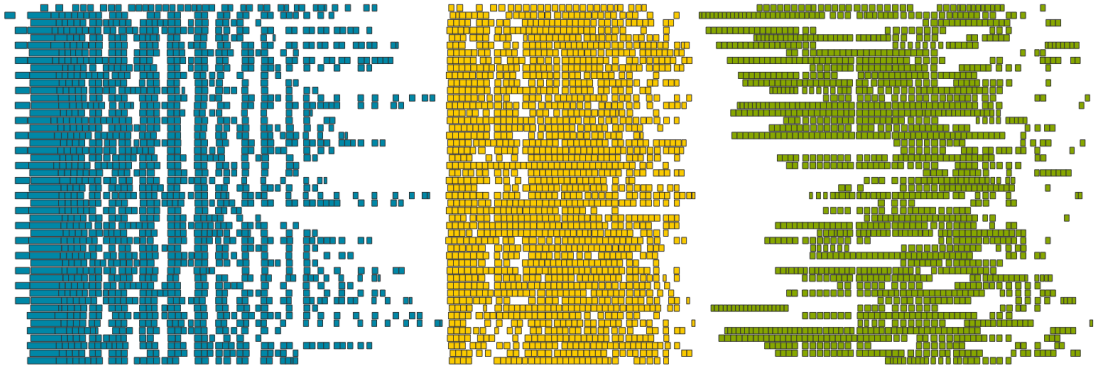
- Many challenges exist for IFS **applications** to run at the Exascale
- First of these is for hardware vendors to build Exascale computers that are both affordable (cost + power) and reliable
- A key consideration is that IFS code must be portable to different architectures (with and without GPGPUs)
- Expectation that 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)
- IFS **applications** (not just the model) will require substantial development in the years to come to run efficiently on Exascale computers

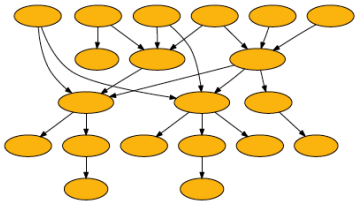
Schedule for future IFS activities in CRESTA

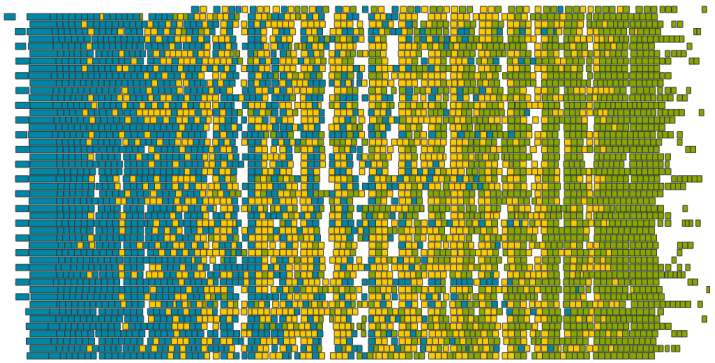
When	Activity
2H13	**Scaling runs of T3999 model on TITAN (CRESTA INCITE award)** **IFS Legendre transform DGEMMs on TITAN GPUs (libsci_acc)**
4Q13-1Q14	**Further IFS scalability optimizations** • Radiation [wave model, surf scheme] computations in parallel with model • transpose SL data (to improve memory scaling and performance) **Investigate initialization cost and propose solution** **Development & testing of alternative local data structures (minimizing communications) for IFS**
2014	**Explore use of DAG parallelization (with OMPSs)** • With a toy code representative of IFS **Implement Coarray teams when new F2013 compiler available**

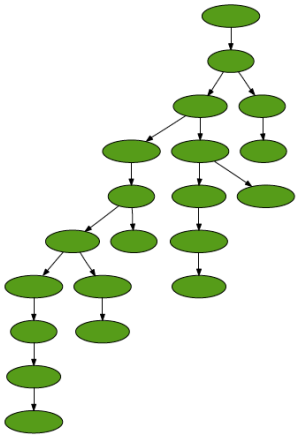
DAG example: Cholesky Inversion

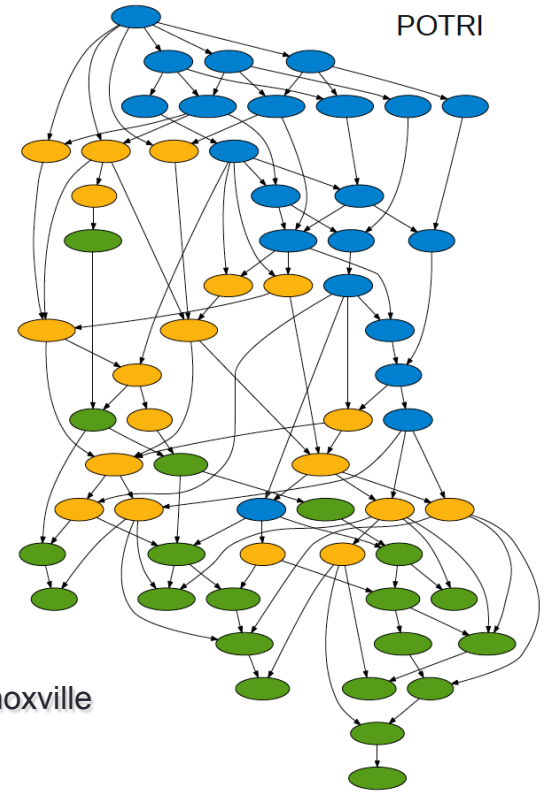












DAG = Directed Acyclic Graph

Can IFS use this technology?

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

Scalability projects at ECMWF (non-CRESTA)

- Explore New Dynamical Core, 2013-2020
  - Alternatives to spectral transform scheme
- Object Oriented Prediction System (OOPS), 2011-2015
  - New 4D-Var data assimilation system with thin controlling C++ layer
  - Single executable instance for 4D-Var trajectory and minimization phases
  - Required for new algorithmic developments, e.g.
    - Parallelization over minimization time window
    - Long window 4D-Var
- Continuous Observation Processing (COPE), 2012-2014
- Benchmark IFS port to GPU architectures, 2014-2016
  - Xeon Phi (OpenMP)
    - RAPS13 T255 model already ported by CSC, on 3 knc MIC cards
  - NVIDIA (OpenACC)

Conclusions

- CRESTA's focus is to work on the software challenge posed by exascale
- This challenge is driven by the likely hardware solutions that will be developed to deliver an exaflop
- Unprecedented levels of heterogeneous parallelism mean we need to think about:
  - What are the scientific challenges that need this level of computational power and complexity?
  - What new modelling and algorithmic methods do we need?
  - How do we keep 100+ million threads busy?
- Hardware and software are inextricably linked
- Some of the software challenges CRESTA sees can only be solved by rethinking the way we compute in parallel

CRESTA