



CRESTA OVERVIEW

Professor Mark Parsons

Project Coordinator
EPCC Executive Director

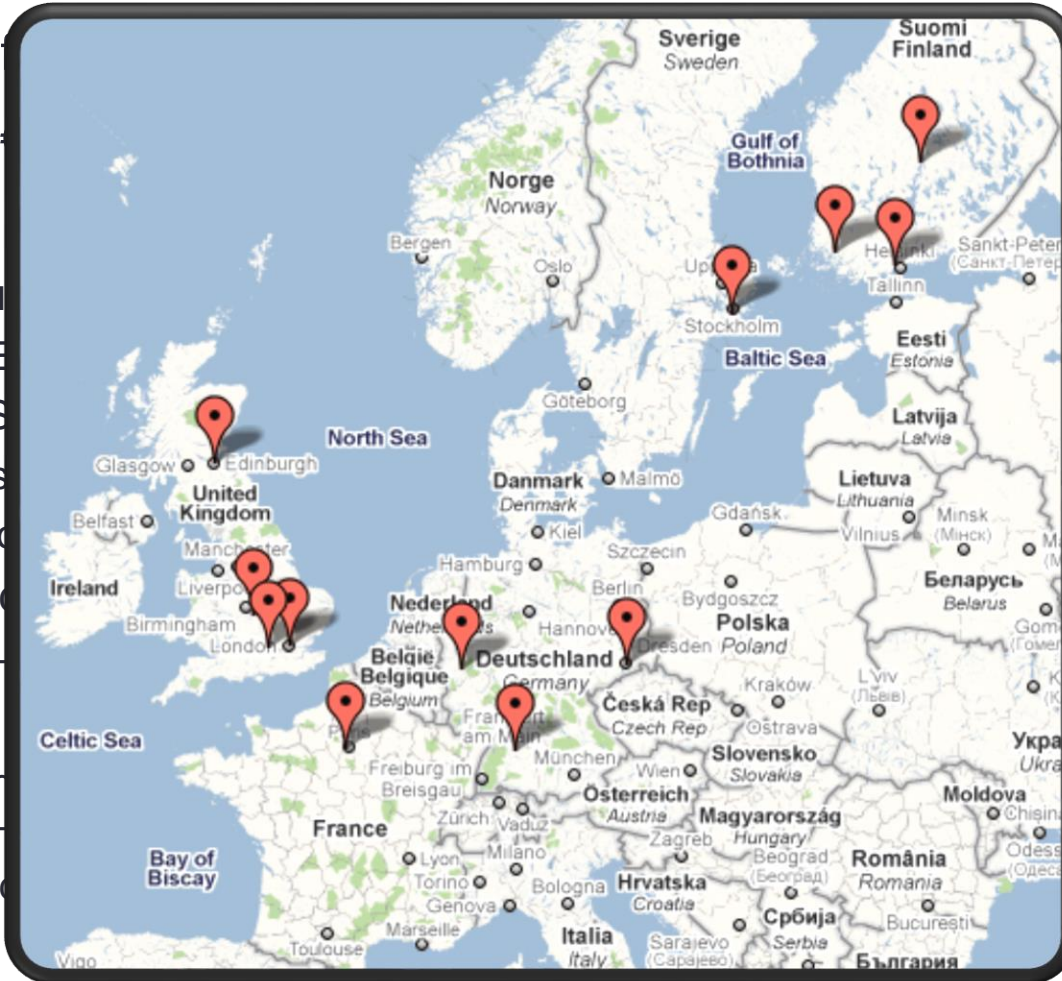


Collaborative Research into Exascale Systemware, Tools and Applications - CRESTA

- Exascale Software Co-design
- 39 months - €12million
- Leading European HPC centres
 - EPCC – Edinburgh, UK
 - HLRS – Stuttgart, Germany
 - CSC – Espoo, Finland
 - KTH – Stockholm, Sweden
- A world leading vendor
 - Cray UK – Reading, UK
- World leading tools providers
 - Technische Universitaet Dresden (Vampir) – Dresden, Germany
 - Allinea Ltd (DDT) – Warwick, UK
- Coordinated by EPCC
- Ended 31st December 2014
- Exascale application owners and specialists
 - Abo Akademi University – Abo, Finland
 - Jyvaskylan Yliopisto – Jyvaskyla, Finland
 - University College London –UK
 - ECMWF – Reading, UK
 - Ecole Central Paris – Paris, France
 - DLR – Cologne, Germany
 - KTH – Stockholm, Sweden
 - USTUTT – Stuttgart, Germany

Collaborative Research into Exascale Systemware, Tools and Applications - CRESTA

- Exascale So...
- 39 months -
- Leading Eu...
 - EPCC – B...
 - HLRS – S...
 - CSC – Es...
 - KTH – Sto...
- A world lea...
 - Cray UK –
- World leadi...
 - Technisch... (Vampir) –
 - Allinea Lt...



2014

owners and

ity – Abo, Finland
Jyvaskyla,

ndon –UK

UK

Paris, France

any

eden

Germany

A hardware *and* software challenge

- 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 leader scales to 92 million cores and 526MW at the Exascale
- Slower better balanced cores mean parallelism at the 500 million – 1 billion thread scale

A hardware *and* software challenge

- 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 accelerators can't get to 20MW
- The Exascale exponential growth in data flow and memory hierarchies inside the node will not be well balanced compared to the number of cores
- The solution will be to use more cores per node with better communications
- But these solutions **INC**rease the need for parallelism
 - Today's leader scales to 92 million cores and 526MW at the Exascale
- Slower better balanced cores mean parallelism at the 500 million – 1 billion thread scale

Hardware is leaving software behind – we don't know how to use such parallelism

At the Exascale software leaves algorithms behind

- Few mathematical algorithms are designed with parallelism in mind
 - ... parallelism is “just a matter of implementation” is the mind-set
- This approach generates much duplication of effort as components are custom-built for each application
 - ... but the years of development and debugging inhibits change and users are reluctant to risk a reduction in scientific output while rewriting takes place
- HPC is at a “tipping point”
 - Without fundamental algorithmic changes progress in many areas will be limited ... and not justify the investment in Exascale systems
 - But it's not just a case of re-writing applications – it's much more difficult
- This doesn't just apply to Exascale
 - It's apparent today at the Petascale ... as most people know
- And we have a huge skills and tools gap ...

At the Exascale software leaves algorithms behind

- Few mathematical algorithms are designed with parallelism in mind
 - ... parallelism is “just a matter of implementation” is the mind-set
- This approach generates much debt that if components are custom-built for each
 - ... but the years of effort and users are reluctant to risk taking place
- HPC is at a “tipping point”
 - Without fundamental changes areas will be limited ... and no systems
 - But it’s not just a case of ... much more difficult
- This doesn’t just apply to Exascale
 - It’s apparent today at the Petascale ... as most people know
- And we have a huge skills and tools gap ...

Software and how we
model and simulate
remain the key
Exascale challenges

Key principles behind CRESTA

- Two strand project
 - Building and exploring appropriate *systemware* for Exascale platforms
 - Enabling a set of key *co-design* applications for Exascale
- Co-design was at the heart of the project. Co-design applications:
 - Provided guidance and feedback to the systemware development process
 - Integrated and benefited from this development in a cyclical process
- Employed both incremental and disruptive solutions
 - Exascale requires both approaches
 - Particularly true for applications at the limit of scaling today
 - Solutions have also helped codes scale at the peta- and tera-scales
- Project has been committed to open source for interfaces, standards and new software – has published much of its work as white papers and case studies

Key principles behind CRESTA

- Two strand project
 - Building and exploring appropriate
 - Enabling a set of key *co-design* ap
- Co-design was at the heart of the
 - Provided guidance and feedback to
 - Integrated and benefited from this develop... a cyclical process
- Employed both incremental and disruptive solutions
 - Exascale requires... approaches
 - Particularly true for... the limit of scaling today
 - Solutions
- Project has... and new sc... and case s...
 - 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

Disruptive approach

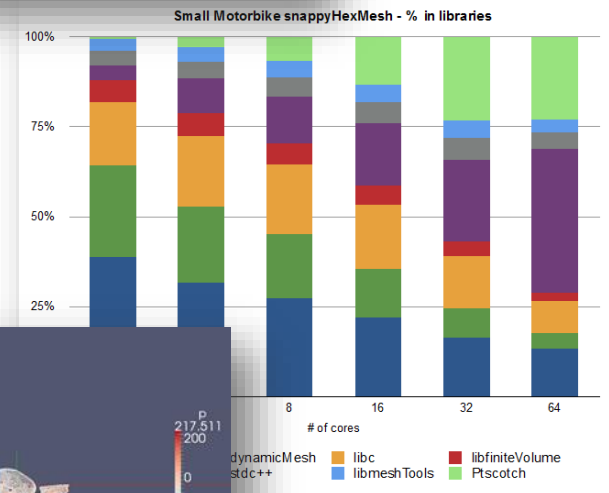
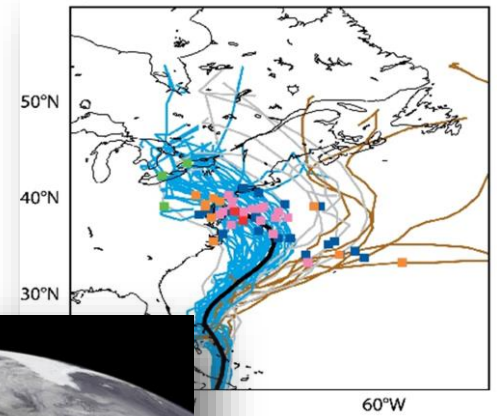
- Work with co-design applications to consider alternative algorithms
- Crucial we understand maximum performance before very major application redesigns undertaken

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

The challenges of disruption

- Major codes are often developed over long time periods – 20+ years
- Disruptive change at their core is:
 - Complex
 - Costly
 - Time consuming
- But for the Exascale, CRESTA showed that it's needed
 - e.g. IFS and OpenFoam
- Getting people to think disruptively is **REALLY DIFFICULT**
 - We're all bad at forgetting what we 'know' is impossible



CRESTA and Exascale hardware

- CRESTA's model of software co-design has been criticised as 'not proper co-design' at times
- But you don't need to build hardware to use its parameters to inform your co-design process
- When working we assumed
 - Massive parallelism – 100+ million threads
 - Probably heterogeneous threads e.g. CPU + accelerator
 - Complex multi-layer memory and I/O hierarchies
 - Non-uniform network topologies and performance
- Our work has also informed hardware design through our HPC vendor partner – Cray – and other vendors who read publications / attend talks etc

Some comments on hardware developments

- We are beginning to see how vendors are approaching the Exascale
- A key period will be 2017 – 2019 where a range of new technical solutions will come onto the market
 - We are currently in the ‘lull before the storm’
- Some contend that CRESTA is too negative about thread counts
- A key challenge is turning processor and node-level developments into systems at scale
- Few vendors exist today who can build reliable large systems
 - Just producing node-level technologies will not be enough
- There is a real need for much greater engagement between applications, systemware and hardware designers

Where are we on the road to Exascale?

- The call text from 2010 said:

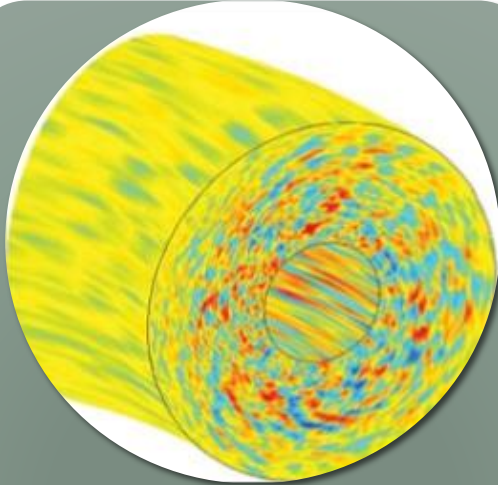
The target is to develop a small number of advanced computing platforms with potential for extreme performance (100 petaflop/s in 2014 with potential for exascale by 2020), as well as optimised application codes driven by the computational needs of science and engineering and of today's grand challenges such as climate change, energy, industrial design and manufacturing, systems biology. These platforms should rely on vendors' proprietary hardware or on COTS.

- The fastest machine in Europe is 6.3 Petaflops Linpack in 2015
- The fastest machine in the World is 33.9 Petaflops Linpack in 2015

Power, parallelism and planning – the challenges

- **POWER:** the only country with a published, published plan for ‘close to’ Exascale is Japan – and it is targeting 2020
 - But it is clear that the first Exascale systems will have a power requirement of around 50-60 MW
 - At EPCC’s current price that’s an annual electricity bill of ~ €70 million
 - For EPCC it’s difficult to see how we could physically host a system requiring more than 20 MW
- **PARALLELISM:** as CRESTA has shown parallelism at the Exascale is still **THE** core challenge for systemware, software and algorithms
 - Clear we need much more algorithmic innovation
- **PLANNING:** a long-term view is crucial
 - Changes to codes with big user communities will not come overnight
 - There is considerable expense associated – Exascale must be about investment in software as well as hardware

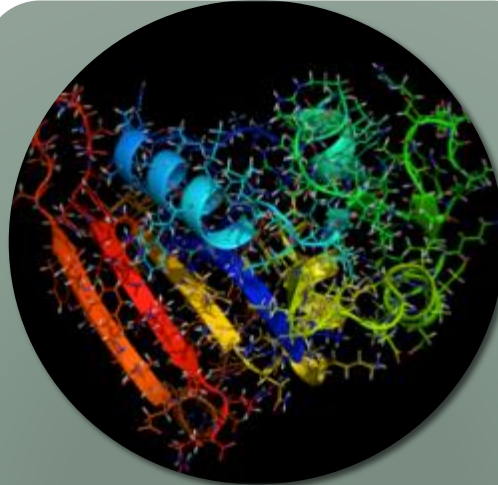
Co-design applications



Elmfire

Gyrokinetic code for turbulent fusion plasma
 Simulating plasma behavior in large scale fusion reactors

An almost complete code restructuring
 Radical reduction of memory consumption per core



GROMACS

Molecular dynamics for

- Modelling of biological systems
- computational material and drug design

10M atom simulation

Coupling strong scaling techniques with ensemble scaling

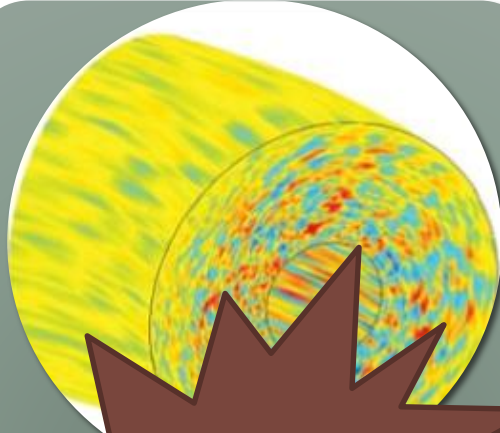


HemeLB

Simulation of cerebrovascular bloodflow, using LB
 Medical simulations to help surgeries
 Brain aneurysm simulation

Pre- and post-processing and load balancing
 Hybridisation, restructuring

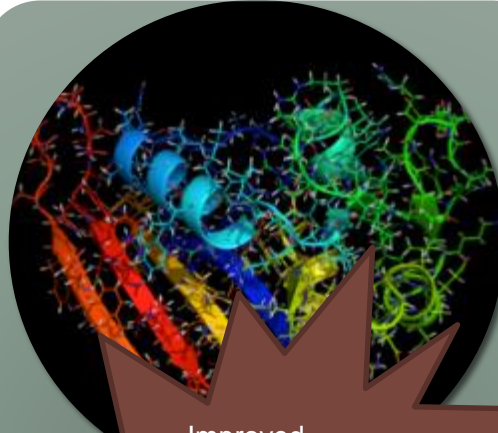
Co-design applications



Exascale 3D
decomposition
and
visualisation

Elm
Gy
fur
Simul
plasma behavior in
large scale fusion reactors

An almost complete code
restructuring
Radical reduction of memory
consumption per core



Improved
implementations
+ code reorg for
task parallelism
+ ensemble
engine

• Computational material
and drug design
10M atom simulation

Coupling strong scaling
techniques with ensemble
scaling



Physics for
Exascale +
performance /
scaling of LB

Medical simulations to help
surgeries
Brain aneurysm simulation

Pre- and post-processing and
load balancing
Hybridisation, restructuring

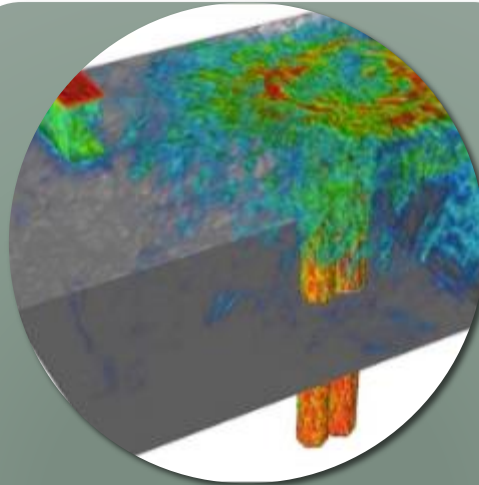
Co-design applications



IFS

Numerical weather prediction
Timely and accurate weather forecasts can save lives
Simulating the trajectory of hurricane Sandy

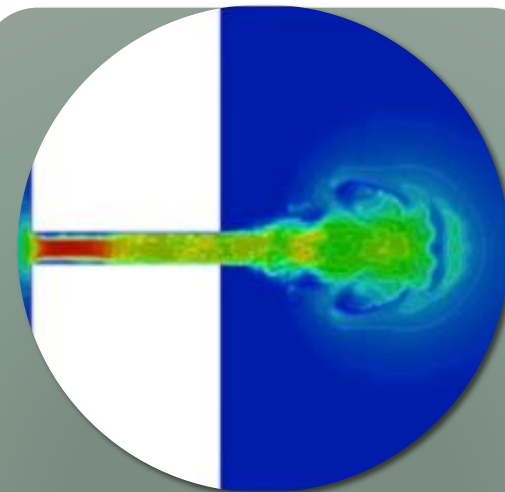
Acceleration
Task-graph based parallelization
New communication models



Nek5000

Open-source CFD
Scaled to 1M cores on Mira
Nuclear power plant cooling simulations

Adaptive mesh refinement
acceleration

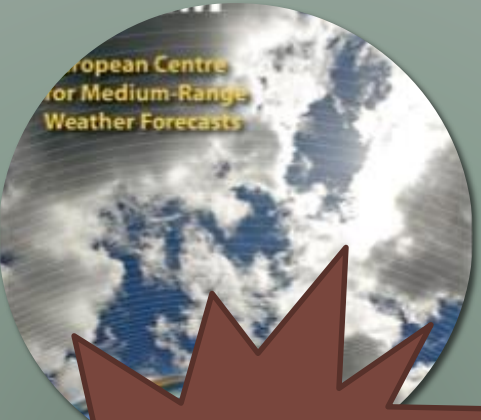
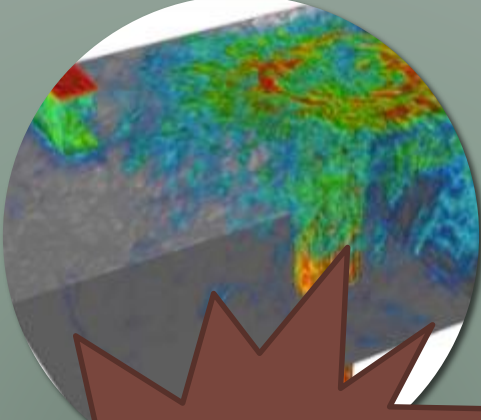
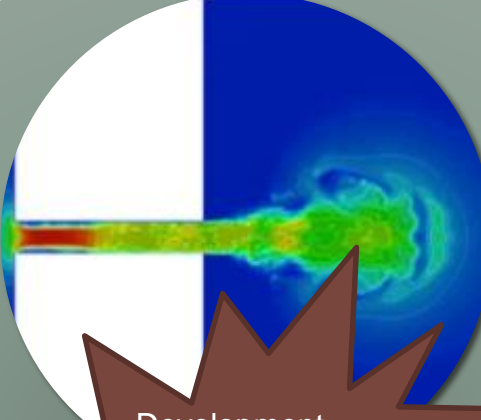


OpenFOAM

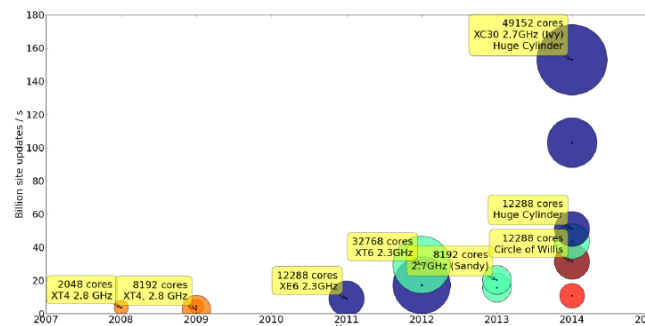
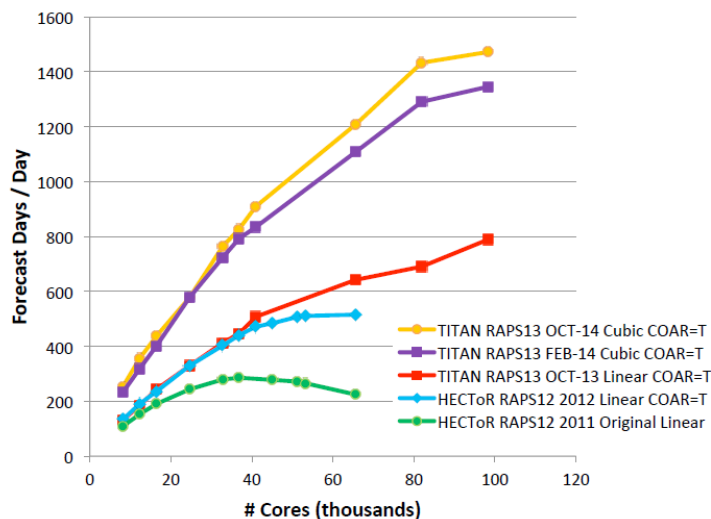
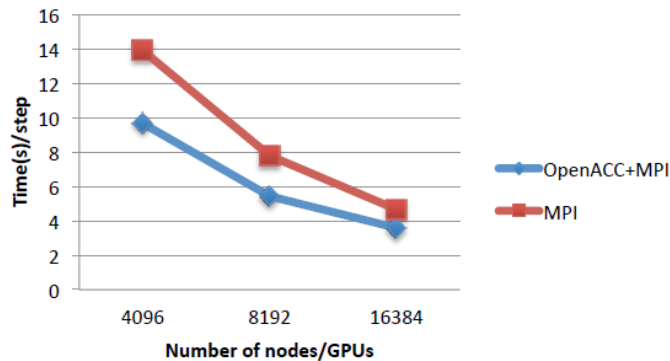
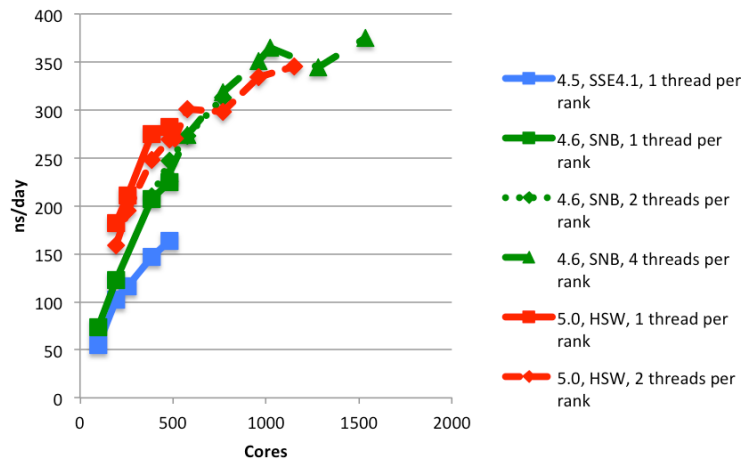
Open-source CFD
Turbulence models: fine resolution in space and time
Wind turbines, hydroelectric power plants
Francis pump turbine simulation

Linear solver optimization

Co-design applications

 <p>PGAS approaches + cubic grid + OmpSs experiments</p> <p>IES</p> <p>Simulation of the trajectory of hurricane Sandy</p> <p>Acceleration Task-graph based parallelization New communication models</p>	 <p>GPGPU engine + AMR + Exascale mesh partitioner</p> <p>Simulation of plant cooling</p> <p>Adaptive mesh refinement acceleration</p>	 <p>Development stopped – OpenFOAM is NOT an Exascale code</p> <p>Francis pump turbine simulation</p> <p>Linear solver optimization</p>
---	--	---

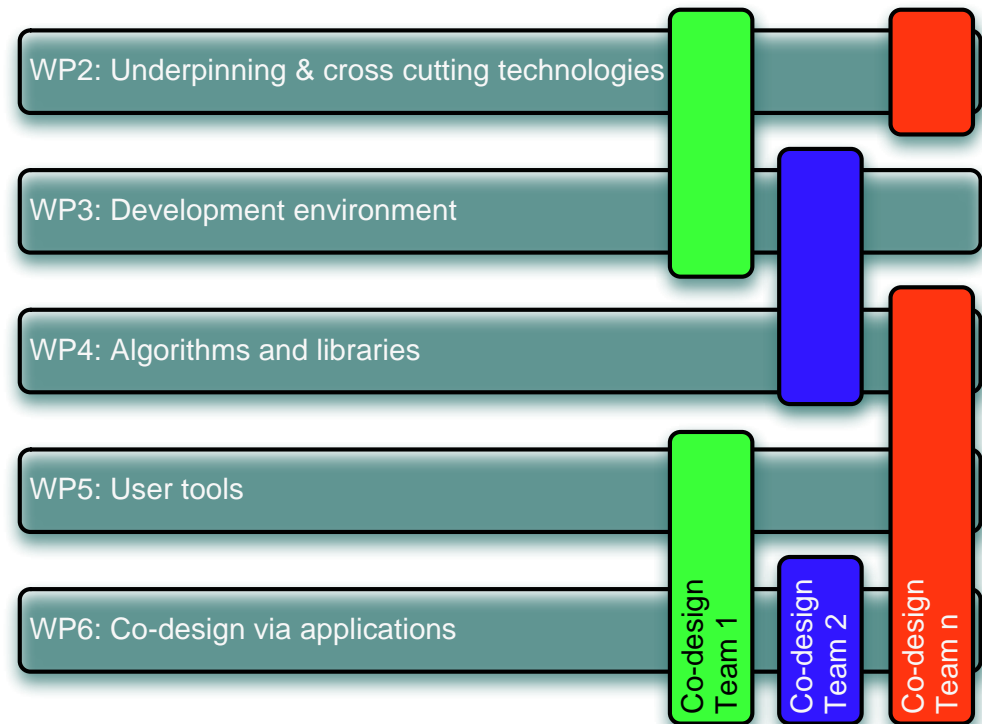
What has CRESTA achieved?



- Lots of scaling results
- But that wasn't really the point of CRESTA

Enabling and managing co-design

- CRESTA thought hard about how to enable and coordinate co-design within the project
- Crucial it got this right
- Generally work packages only encourage 1D collaboration
- Co-design in CRESTA was 2D
- We wanted to work across work packages on specific well-defined challenges
- Very hard to manage but we made it work



Understanding the challenge we face

- We'll show today how software co-design can work
 - Driven by a general understanding of the scale of parallelism that Exascale hardware will deliver
- Identified many challenges – not just with parallelism but also I/O performance, tools, libraries – software and systemware
- Made tools advances which also benefit Petascale
- Shown that some codes e.g. OpenFoam will never run at the Exascale in their present form
- Given code owners the space to explore the Exascale and plan how to respond to it
- A key success has been to create awareness of the challenges – so that management can properly plan and resource

What next?

- CRESTA has stimulated a wide collection of follow-on projects including:
 - EPIGRAM, "Grids in Grids", SimPhoNy, DASH, Introducing Thread and Instruction Level Parallelism into Ludwig, GROMEX, SkaSim, ELP, Score-E, COLOC, ExaFLOW, BioExcel, ComPat, INTERTWinE, NEXTGenIO
- Many of these projects are building on CRESTA activities or are addressing areas CRESTA couldn't look at
- The recently announced FETHPC projects will take forward what CRESTA has learned about how to implement software co-design
- A key challenge that CRESTA identified was around Exascale IO and the NEXTGenIO project will look at this in detail

NEXTGenIO

- CRESTA repeatedly identified IO as a major challenge for Exascale
- Over the next couple of years, new technologies are coming to the data centre building on high performance NV-RAM technologies
- EPCC leads the €8m NEXTGenIO project where
 - Fujitsu and Intel will develop a new HPC platform using latest technology
 - Software strategies for using NV-RAM in HPC systems will be explored
 - These strategies will be tested against real-world models of data centre IO
- These hardware developments have the potential to profoundly change the utilisation of HPC in the data centre
 - HPC may become much more data-centric
- But all of this relies on understanding how best to expose the hardware to the software and vice versa

Conclusions

- CRESTA has shown the scale of the Exascale application challenge
- It has moved some codes towards this challenge and published the results for others to learn from
- Far too few projects like CRESTA exist
 - But the recently announced FETHPC projects are a good step forward!
- It is clear that many instances of DISRUPTIVE INNOVATION will be needed to model and simulate on Exascale systems
- We still need to re-think many algorithms – and continue to build engagement with the mathematics community
- There is a clear need for more software investment
- We need to build the software so the hardware can utilise it!