Software co-design on the road to exascale

Dr Stephen Booth

EPCC Principal Architect

Dr Mark Parsons

EPCC Executive Director
Associate Dean for e-Research

The University of Edinburgh

Overview

It's the end of the world as we know it …

R.E.M. from the album

Document

- Where are we today on the road to exascale?
- Why is exascale such a challenge?
- What is the CRESTA project doing to help solve it

Exascale

- Supercomputers have traditionally undergone geometric growth in performance.
  - Need only look at graphs of the top-500 to see how robust this growth is.
  - Though partly driven by incremental technology developments like process-shrink it is also a self fulfilling prophecy.
    - Industry and consumers expect this growth and do whatever it takes to maintain it.
- Current fastest system is the Sequoia BlueGene/Q @ 20 Pflops
  - First Exascale system expected approx 2020
- Note we say Exascale not Exaflop
  - Total flop rate is just one parameter for a successful machine.
  - Probably not even the most important parameter though it is the easiest to understand.
  - There are many challenges that need to be overcome.

CREST

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.
- European Commission funded project
  - FP7 project
  - Projects started 1st October 2011, three year project
  - 13 partners, EPCC project coordinator
  - €12 million costs, €8.57 million funding

www.cresta-project.eu

CRESTA

Partnership

- Consortium
  - Leading E owners and
    - EPCC –
    - HLRS – rsity – Abo,
    - CSC – E
    - PDC – S – Jyvaskyla,
  - A world le ondon –
    - Cray UK
  - World lead UK
    - Technisc – Paris, France (Vampir) rmany
    - Allinea L

CREST

Motivation behind CRESTA

- We are at a complex juncture in the history of supercomputing
- For the past 20 years supercomputing has “hitched a lift” on the microprocessor revolution driven by the PC
- Hardware has been surprisingly stable
- EPCC in 1994 had the 512 processor Cray T3D system
  - 0.0768 TFlops peak
- EPCC in 2010 retired the 2,560 processor IBM HPCx system
  - 15.36 TFlops peak – 200 x faster but only 5 x more processors ...
- The programming models for these systems were very similar
- But today’s systems present a real problem … which the exascale cruelly exposes

What are the challenges?

- DARPA conducted a study on exascale hardware in 2007
  - Work has been continued by the International Exascale Software Project and, most recently, by CRESTA's first deliverables
- Objective: understand the course of mainstream technology and determine the primary challenges to reaching 1 Exaflop by 2015, or soon thereafter
- They concluded the four key challenges were:
  1. Power consumption
  2. Memory and storage
  3. Application scalability
  4. Resiliency
- See
  - http://www.cse.nd.edu/Reports/2008/TR-2008-13.pdf

CRESTA

1: The power problem

- The most power-efficient microprocessors available today deliver ~600 Mflops/W on Linpack
  - XE6 is ~2.2 MW per petaflop/s … or 2.2GW per exaflop/s
- … clearly, we have to do better!
  - DARPA goal: 50 Gflops/W
  - 100x improvement
- But even then
  - That still equates to a 20MW computer
  - A number of US labs are currently putting in 30-40MW machine room power supplies
- The simplest way to reduce power is to reduce the clock rate … problem for us!

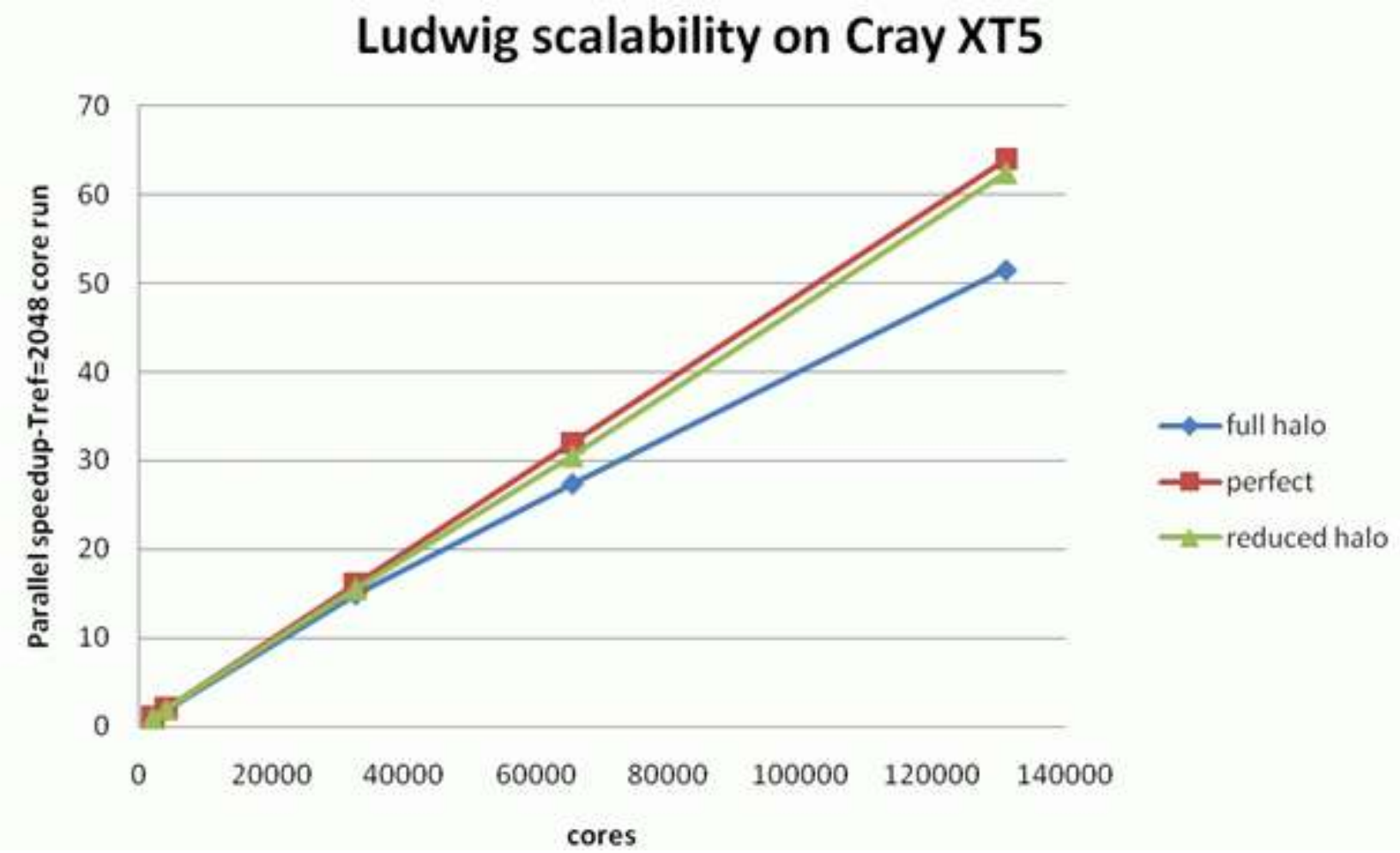
Longannet power station: 2.4 GW

CREST

Parallel computing today

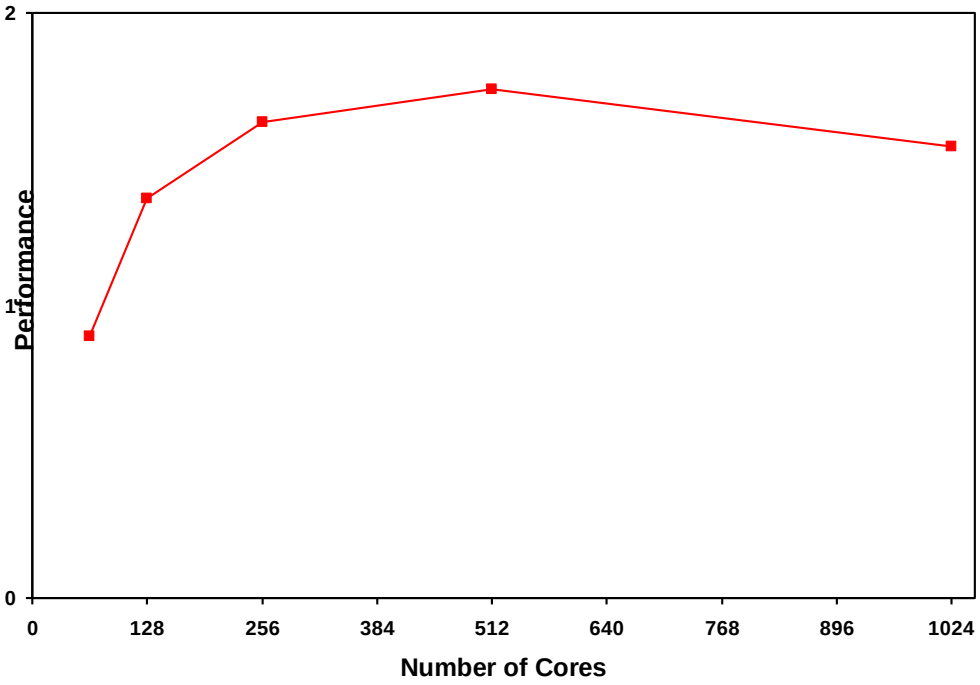
- The programming model is one of a set distinct memories distributed over homogeneous microprocessors
  - Each microprocessor runs a Unix-like OS
- Data transfers between the processors are managed explicitly by the application
- Almost all programs are written in sequential Fortran or C
- They use MPI (Message Passing Interface) for data transfers between nodes/microprocessors
- Some applications which exploit parallel threads on each microprocessor use the hybrid model
  - Shared memory on the microprocessor, distributed memory beyond
  - This holds promise for many applications, but is still rare

Scaling to very large core counts is possible …



… but often is not

- For example this typical chemistry code



- This behaviour is caused by the overheads of global communication

Hardware is leaving software behind

- Hardware is leaving many HPC users and codes behind
- Majority of codes scale to less than 512 cores
  - These will soon be desktop systems
- Less than 10 codes in EU today will scale on capability systems with 100,000+ cores
  - HECToR service already has 90,112 cores
  - Germany's Jugene system already has 294,912 cores
- Many industrial codes scale very poorly – some codes will soon find a laptop processor a challenge!
- Much hope is pinned on accelerator technology
  - But this has its own set of parallelism and programming challenges
  - Many porting projects to GPGPU have taken *much* longer than expected
  - Part of the solution but not a magic bullet.

Software is leaving algorithms behind

- (Like the OS) few mathematical algorithms have been designed with parallelism in mind
  - … the parallelism is then "just a matter of implementation"
- 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
- Strongly believe we are at a "tipping point"
  - Without fundamental algorithmic changes progress in many areas will be limited … and not justify the investment in exascale systems
- This doesn't just apply to exascale
  - Some codes already fail to scale on an 8 or 16-core desktop system
- And we have a huge skills gap …

DARPA 2007 Aggressive Silicon Strawman

Characteristic	
Flops – peak (PF)	997
- microprocessors	223,872
- cores/microprocessor	742
Cache (TB)	37.2
DRAM (PB)	3.58
Total power (MW)	66.0
Memory bandwidth (B/s per flops)	0.0025
Network bandwidth (B/s per flops)	0.0008

166 million cores !!!

CREST

1. Do SOC designs solve the power problem?

- System-On-a-Chip (SOC) designs provide excellent power savings
  - For example processors and GPUs on a single silicon die
    - AMD's recent APUs for the laptop/netbook market
    - ARM-based tablet processors
- AMD have recently purchased Sea-Micro while Intel have recently purchased the Cray interconnect business
- Almost certainly both vendors intend to embed network hardware on their ever-expanding silicon real estate
  - This makes sense particularly from a power point of view
- At the same time the integration of silicon photonics onto processor dies will happen
  - Certainly all long distance communications will have to be optical
- SOC designs will be key to solving part of the hardware power story

2: Memory and power

- Memory bandwidth has increased ~10x over the past decade
- The energy cost/bit transferred has declined by 2.5x
- … energy cost of driving the memory at full bandwidth has risen 4x
- Memory DIMMs can't provide bandwidth at acceptable energy costs
- And today's applications use more memory than ever before

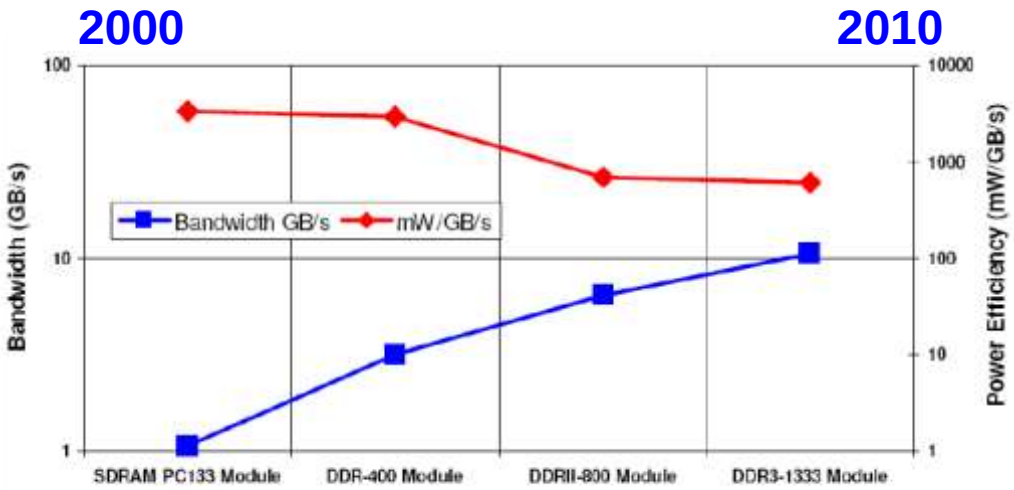
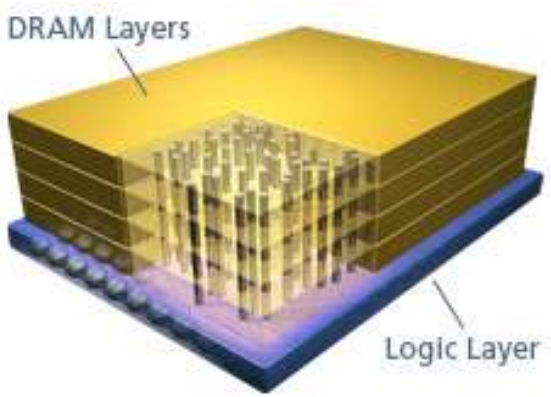


Figure 6.22: Commodity DRAM module power efficiency as a function of bandwidth.

CREST

2: Memory performance

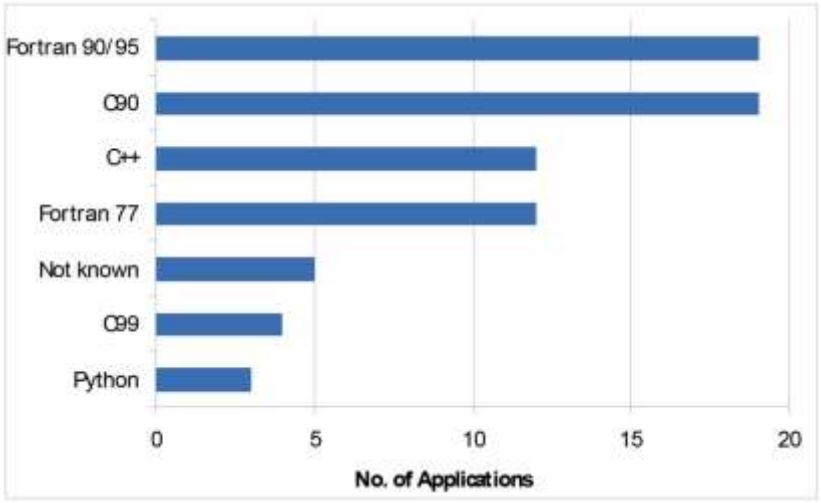
- Over the past 30 years DRAM density has increased ~75x faster than bandwidth
- Memory bandwidth and memory power consumption are the fundamental problem for many exascale system designs
- Multicore processors and accelerators only exacerbate this problem
- Novel memory technologies needed
  - The most likely advance is the introduction of 3D silicon stacking
    - Faster (15X) and more power efficient (70%)
  - More esoteric advances include
    - Faster phase-change memory – much more energy efficient)
    - Memristors – interesting but unproven

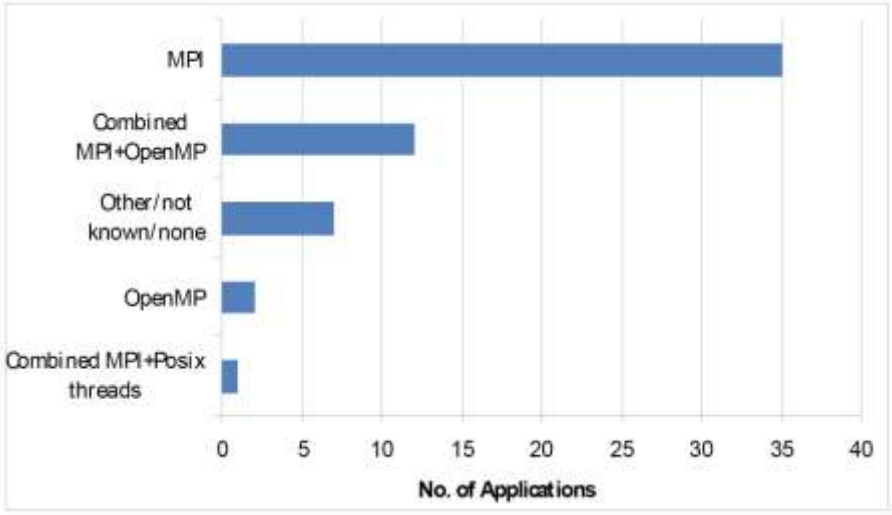


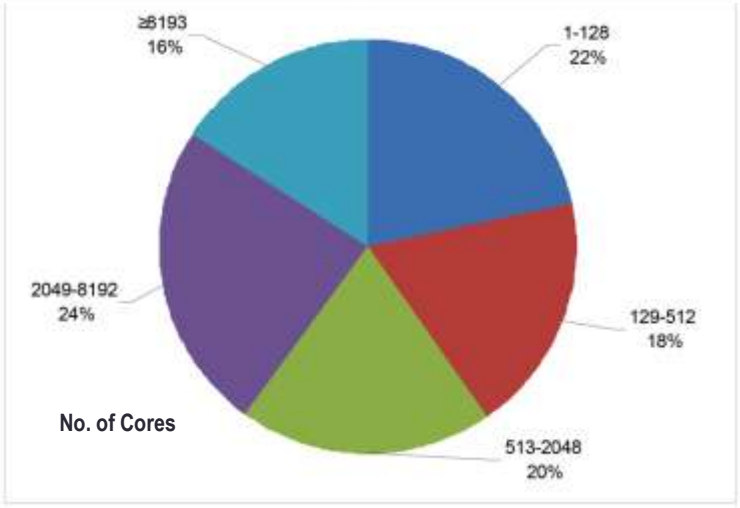
CREST

3. 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

3. Application scalability

- Today's maximum per core performance is 10Gflop/s
  - An Exaflop would therefore require 100 million x86 cores
  - No application today will scale remotely close to this level
- Most codes today use traditional programming models
  - Very little desire by applications community to rewrite using new models
  - But this probably what will be required – most application owners will want to approach major changes incrementally
- Currently taking approach of "*Offer me solutions, offer me alternatives and I decline*" (thanks to R.E.M. for their insight again)
- Performance monitoring and debugging tools - another huge area
  - How do you debug 100 million threads?
  - We're thinking about this in CRESTA
- Also thinking about pre- and post-processing needs at exascale

CRESTA

3. Applications scalability

- Strong versus weak scaling
  - Weak scaling (problem size varies with machine concurrency) has been the mainstay of parallelism for 30 years
  - Strong scaling (scaling with a fixed problem size) has been hard to find
- For some applications there is no more weak scaling because the system being studied is already large enough
  - Example: classical molecular dynamics for many chemistry applications only requires 100 - 1000 atoms/molecules
- An even larger set is constrained by algorithmic complexity
  - There is simply not enough concurrency in the algorithm
  - Modern hardware – multicore and GPGPUs – are cruelly exposing this
- The numerical core (and probably much more) of many applications will have to be rewritten to achieve exascale performance

4: Resiliency

- An exaflop machine may have more than one million processors
  - If each processor has an MTBF of 10 years
    … then the machine will have a MTBF of ~5 minutes!
- We therefore have to be able to operate it in a way which is resilient to single-node failures
  - Or partial problems with other components e.g. the interconnect
- Unfortunately, most scientific applications today use synchronous algorithms
- … which halt when something blocks the data flows
- Fault tolerance is not a new problem
  - von Neumann considered this in detail as early computers failed often
- Much work remains to be done
  - This is an area where hardware and software (particularly systemware) co-design are crucial

Hardware co-design

- All vendors have the same hardware challenges
- It would be possible to build an exascale system today … there's no hardware reason why not
  - Indeed, China announced it will build 2 x 100Pflop systems in next 3 years at the IESP meeting in Japan in April 2012
- But the system will be very difficult to use from a software application point of view … and almost certainly the systemware (OS, compilers, debuggers, etc.) will struggle too
- In CRESTA sees exascale as a SOFTWARE challenge
- We're therefore working from a broad understanding of what exascale hardware will be like and focussing our efforts on software
  - … in this context software is both systemware and applications

CRESTA

Comments on the direction of worldwide HPC

- We need to be very careful that exascale doesn't become an "Apollo Programme" for HPC
- The need for exascale systems must be driven by science and research challenges
- The desire to build exascale systems mustn't obscure *why* we're building these systems
  - Or the long-suffering tax payer will stop the funding
- The balance of spending between software and hardware seems wrong
  - Far too little is spent on applications – particularly new applications of modelling and simulation
  - Without new applications the number of applications that can execute at the top-end of HPC will dwindle to zero
- We are dangerously close to forgetting why we need the exascale

Key principles of CRESTA

- Two strand project
  - Building and exploring appropriate *systemware* for exascale platforms
  - Enabling a set of key *co-design* applications for exascale
- Co-design is at the heart of the project. Co-design applications:
  - provide guidance and feedback to the systemware development process
  - integrate and benefit from this development in a cyclical process
- Employing both incremental and disruptive solutions
  - Exascale requires both approaches
  - Particularly true for applications at the limit of scaling today
  - Solutions will also help codes scale at the peta- and tera-scales
- Developing the exascale software stack
- Committed to open source interfaces, standards and new software

CRESTA

Co-design Applications

- Exceptional group of six applications used by academia and industry to solve critical grand challenge issues
- Applications are either developed in Europe or have a large European user base
- Enabling Europe to be at the forefront of solving world-class science challenges

Application	Grand challenge	Partner responsible
GROMACS	Biomolecular systems	KTH (Sweden)
ELMFIRE	Fusion energy	ABO/ JYU (Finland)
HemeLB	Virtual Physiological Human	UCL (UK)
IFS	Numerical weather prediction	ECMWF (International)
OpenFOAM	Engineering	EPCC / HLRS / ECP
Nek5000	Engineering	KTH (Sweden)

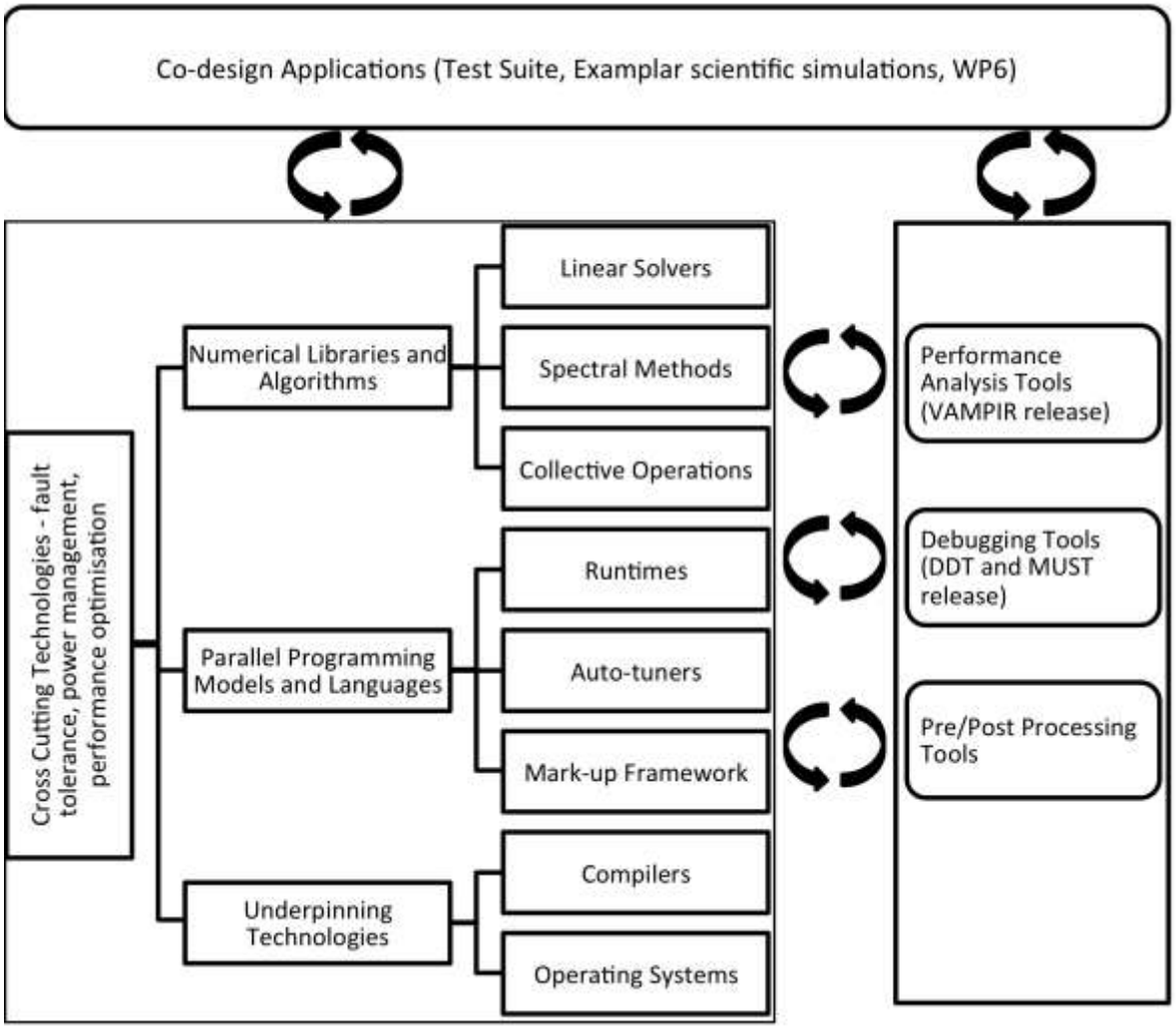
CRESTA

Systemware

- Systemware is the software components required for grand challenge applications to exploit future exascale platforms
- Consists of
  - Underpinning and cross cutting technologies
    - Operating systems, fault tolerance, energy, performance optimisation
  - Development environment
    - Runtime systems, compilers, programming models and languages including domain specific
  - Algorithms and libraries
    - Key numerical algorithms and libraries for exascale
  - Debugging and Application performance tools
    - Very lucky to have world leaders in CRESTA
      - Allinea's DDT, TUD's Vampir and KTH's perfminer
  - Pre- and post- processing of data resulting from simulations
    - Often neglected, hugely important at exascale

CRESTA

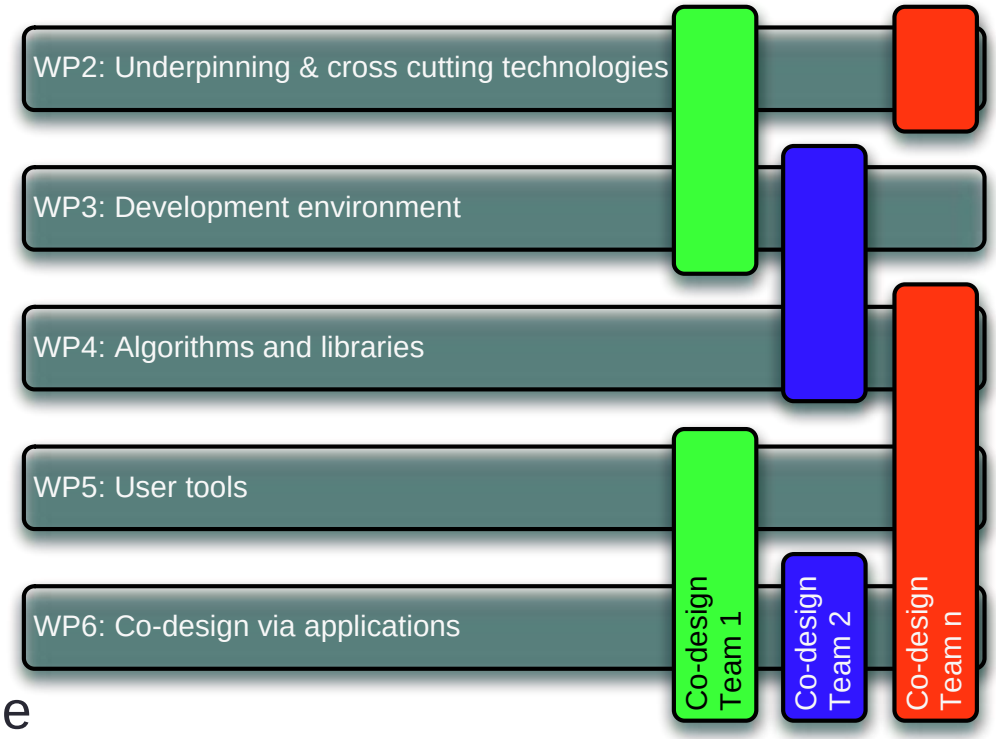
Relationship between activities



CREST

Enabling and managing co-design

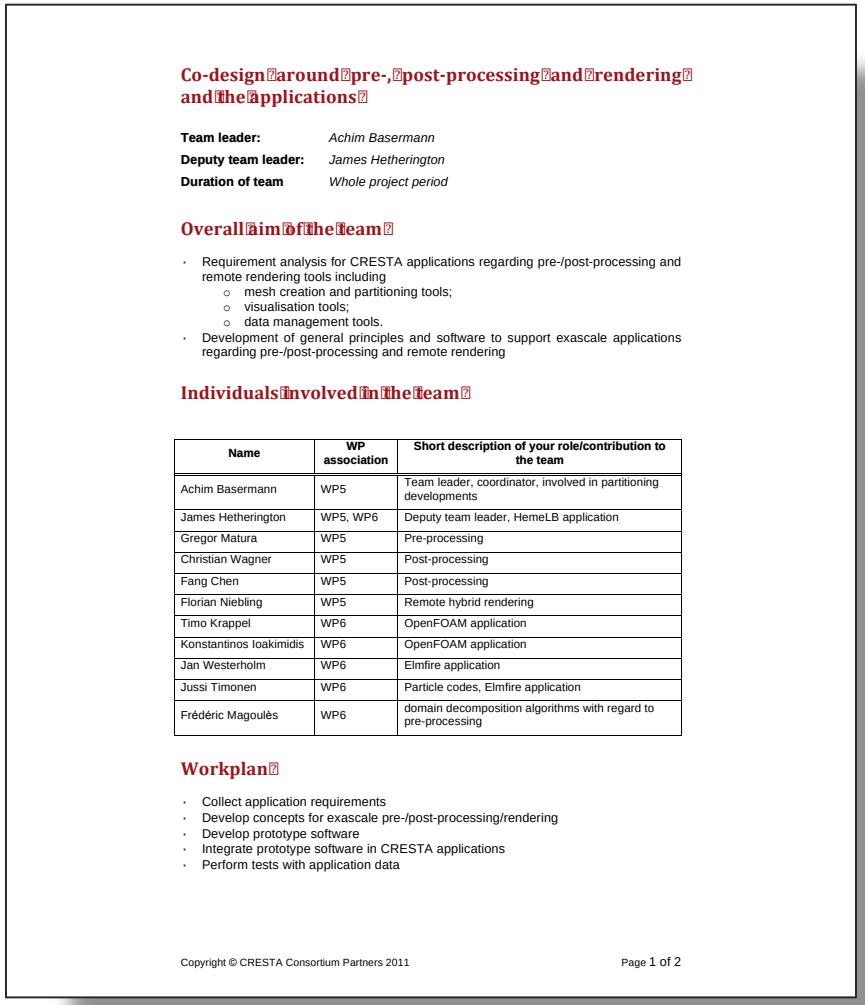
- We have thought hard about how to enable and coordinate co-design within the project
- Crucial we get this right
- But work packages only encourage 1D collaboration
- Co-design in CRESTA is 2D
- We want to work across work packages on specific well-defined challenges
- We want to be able to report the results via the relevant work packages – and learn from them throughout the project



CRESTA

Co-design teams

- These have already been set up.
- Each team has participants from multiple work packages
- There is always at least one application represented (and often several)
- Each team has produced a short document outlining its membership, goals and challenges



Co-design around pre-, post-processing and rendering and the applications

Team leader: *Achim Basermann*
Deputy team leader: *James Hetherington*
Duration of team *Whole project period*

Overall aim of the team

- Requirement analysis for CRESTA applications regarding pre-/post-processing and remote rendering tools including
  - mesh creation and partitioning tools;
  - visualisation tools;
  - data management tools.
- Development of general principles and software to support exascale applications regarding pre-/post-processing and remote rendering

Individuals involved in the team

Name	WP association	Short description of your role/contribution to the team
Achim Basermann	WP5	Team leader, coordinator, involved in partitioning developments
James Hetherington	WP5, WP6	Deputy team leader, HemeLB application
Gregor Matura	WP5	Pre-processing
Christian Wagner	WP5	Post-processing
Fang Chen	WP5	Post-processing
Florian Niebling	WP5	Remote hybrid rendering
Timo Krappel	WP6	OpenFOAM application
Konstantinos Ioakimidis	WP6	OpenFOAM application
Jan Westerholm	WP6	Elmfire application
Jussi Timonen	WP6	Particle codes, Elmfire application
Frédéric Magoulès	WP6	domain decomposition algorithms with regard to pre-processing

Workplan

- Collect application requirements
- Develop concepts for exascale pre-/post-processing/rendering
- Develop prototype software
- Integrate prototype software in CRESTA applications
- Perform tests with application data

Co-design teams

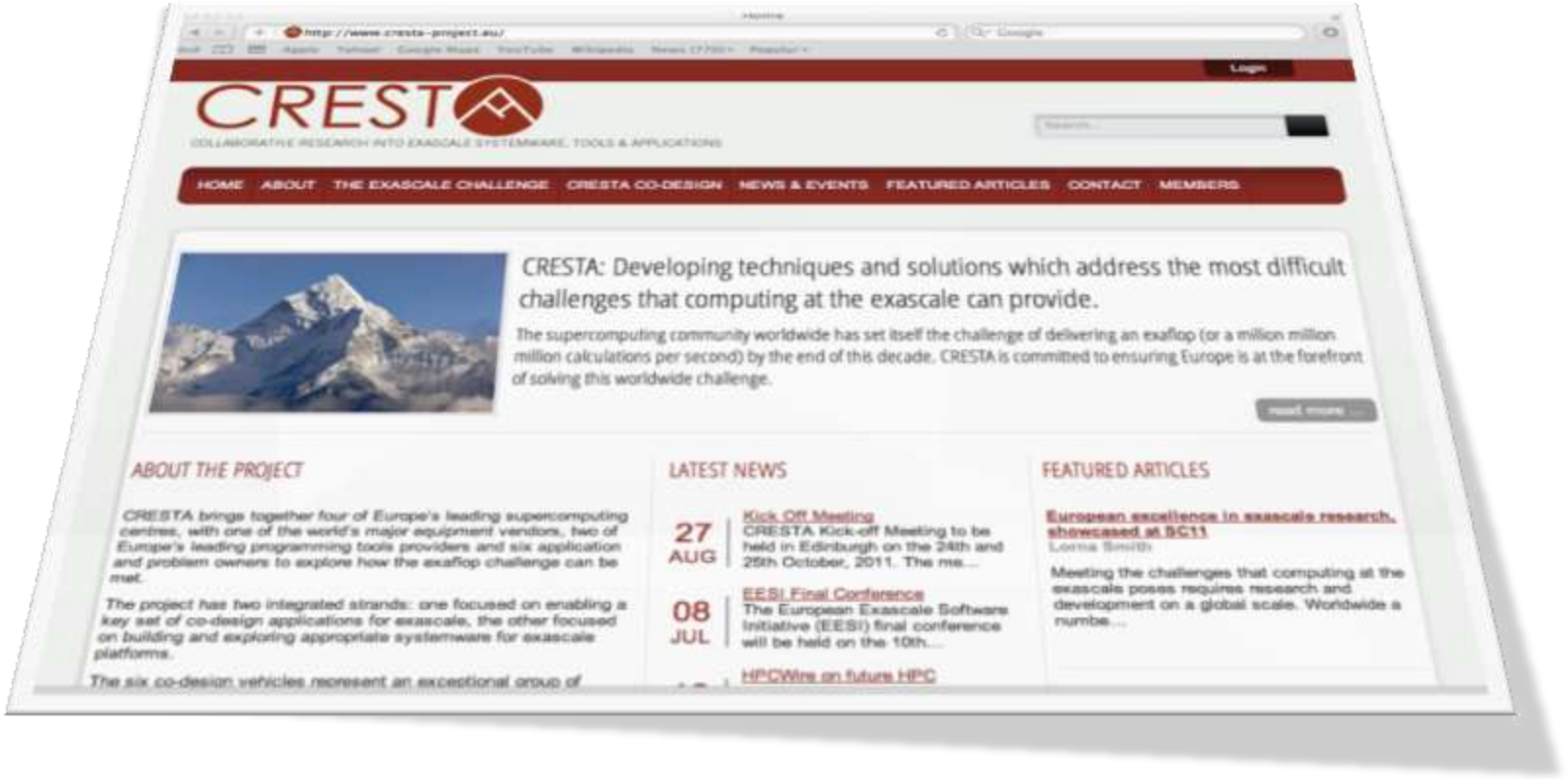
Co-design Team	Team Leader(s)
Global Monitoring for Application Performance Optimization	Michael Schliephake (KTH)
Runtime-Support for Hybrid Parallelisation	Michael Schliephake (KTH)
Development Environment	David Lecomber (Allinea)
Co-array Fortran	George Mozdzynski & Mats Hamrud (ECMWF)
FFT	Stephen Booth & David Henty (EPCC)
Linear Solvers and Pre Conditioners	Dmitry Khabi & Timo Krappel (USTUTT)
Pre-, Post-processing and Rendering and the Applications	Achim Basermann (DLR), James Hetherington (UCL)
GP-GPU	Alan Gray (EPCC)
Lattice Boltzmann	Keijo Mattila (JYU)
Weak-Strong Scaling/Ensemble	Jan Åström, (CSC), Jan Westerholm. (ABO)
Benchmark Suite	Jeremy Nowell (EPCC)

Conclusions

- CRESTA is one “small” project amongst many that are tackling the exascale challenge
- It’s focus on software co-design is probably unique
- Hardware is slowly moving forward – and will probably deliver the first exascale systems in early 2020’s
- Far too little funding is being focussed on the software side (particularly developing previously infeasible simulations)
- CRESTA’s focus on the exascale software stack (both applications and systemware) is trying to redress this balance
- We need to be brave and plan our disruptive work now – not in 2019!

CRESTA

http://www.cresta-project.eu



It's the end of the world as we know it

and I feel fine …

R.E.M.