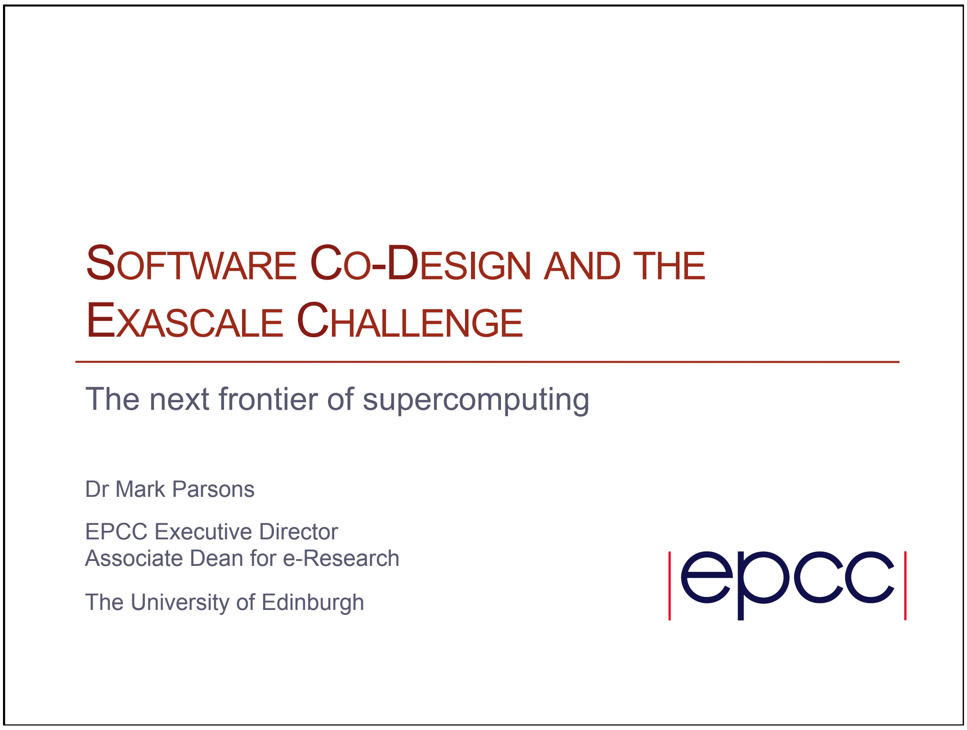


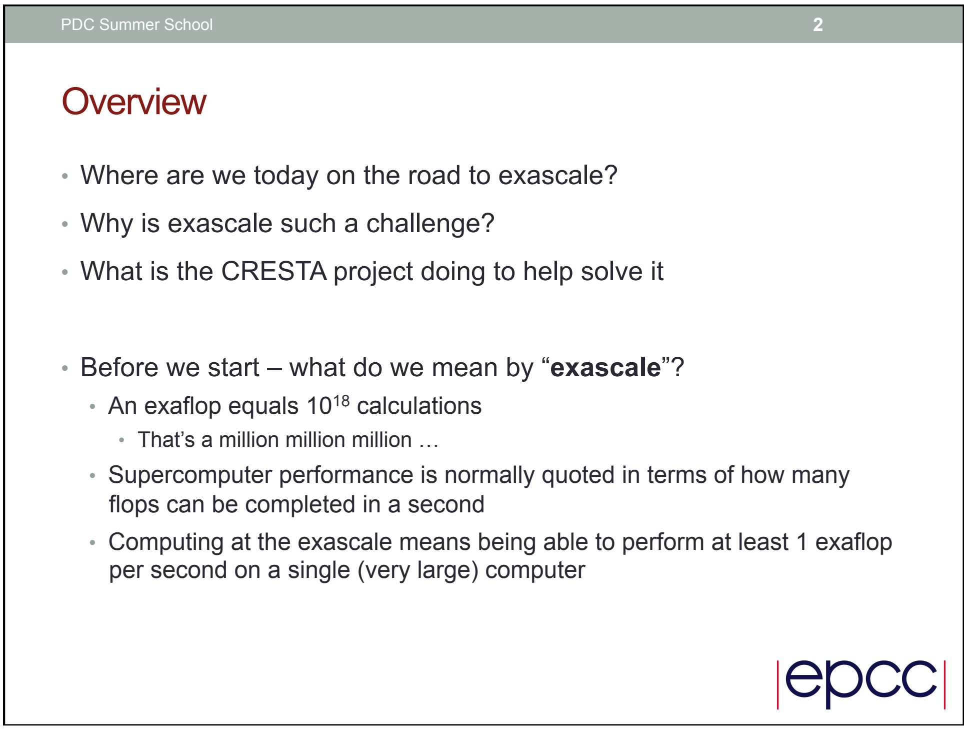
Software Co-Design and the Exascale Challenge

The next frontier of supercomputing

Dr Mark Parsons

EPCC Executive Director
Associate Dean for e-Research

The University of Edinburgh

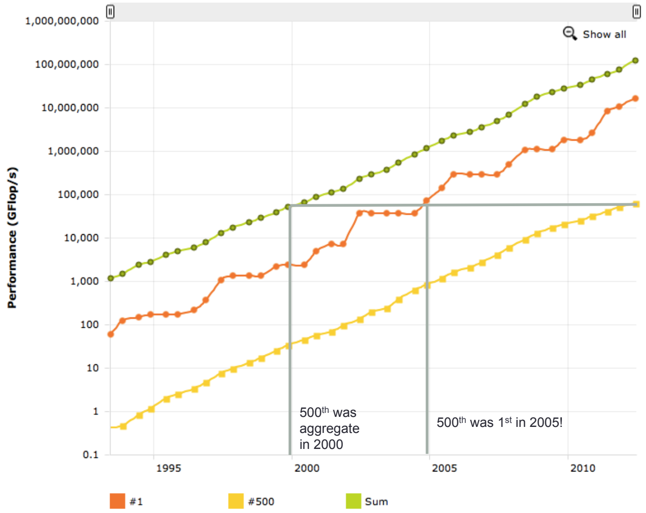


Overview

- 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

- Before we start – what do we mean by “**exascale**”?
  - An exaflop equals 10^{18} calculations
    - That’s a million million million …
  - Supercomputer performance is normally quoted in terms of how many flops can be completed in a second
  - Computing at the exascale means being able to perform at least 1 exaflop per second on a single (very large) computer

Top500 Evolution



500th was aggregate in 2000

500th was 1st in 2005!

Aggregate Top500 performance

1st system

500th system

source: www.top500.org

Top500

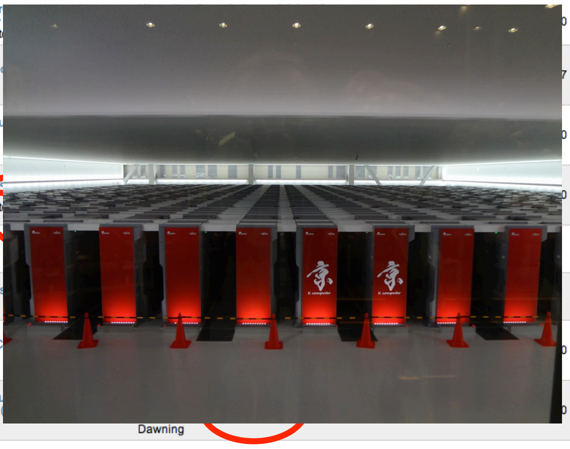
1.57 million cores

16 petaFLOPS

8 MW

Rank	Site	Computer/Year Vendor	Cores	R_{max}	R_{peak}	Power
1	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom / 2011 IBM	1572864	16324.75	20132.66	7890.0
2	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect / 2011 Fujitsu	705024	10510.00	11280.38	12659.9
3	DOE/SC/Ar Laboratory United Stat					0
4	Leibniz Re Germany					7
5	National Su Tianjin China					0
6	DOE/SC/O Laboratory United Stat					0
7	CINECA Italy					
8	Forschungs Germany					
9	CEA/TGCC France					0
10	National Su Shenzhen (China	Dawning				0

GPU



Japan's K computer

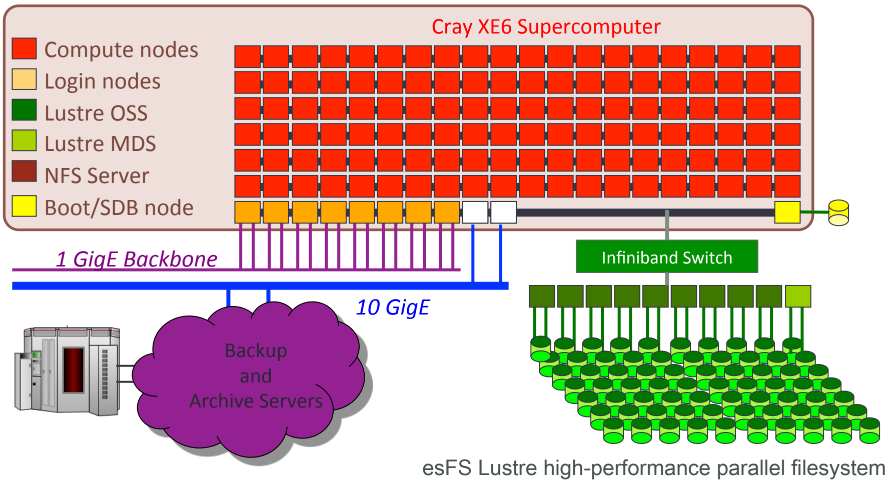
source: www.top500.org

Last update: June 2012

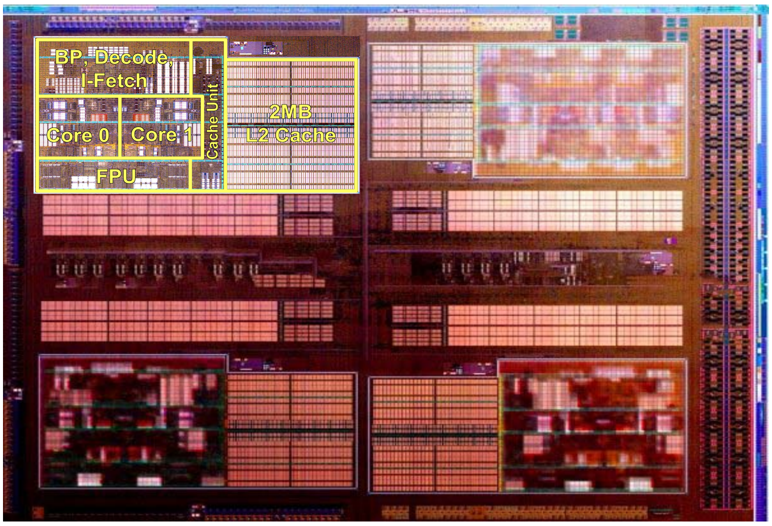
Parallel computing today

- 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

Cray XE6 at Edinburgh

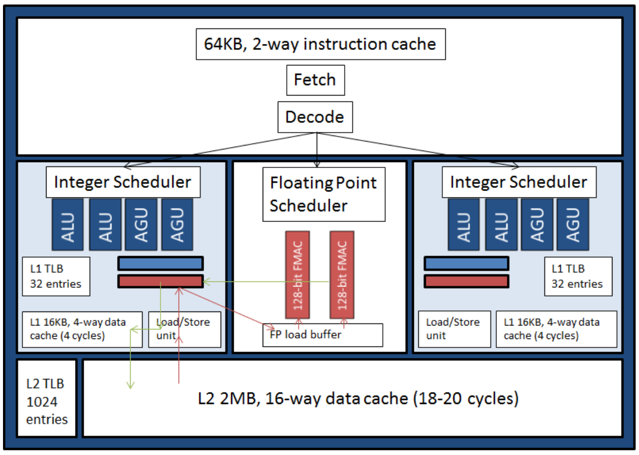


AMD Interlagos Die



Two of these dies make up an Interlagos processor

Interlagos dual bulldozer-core module



64KB, 2-way instruction cache

Fetch

Decode

Integer Scheduler

ALU ALU AGU AGU

L1 TLB 32 entries

L1 16KB, 4-way data cache (4 cycles)

Load/Store unit

Floating Point Scheduler

128-bit FMAC

128-bit FMAC

FP load buffer

Integer Scheduler

ALU ALU AGU AGU

L1 TLB 32 entries

Load/Store unit

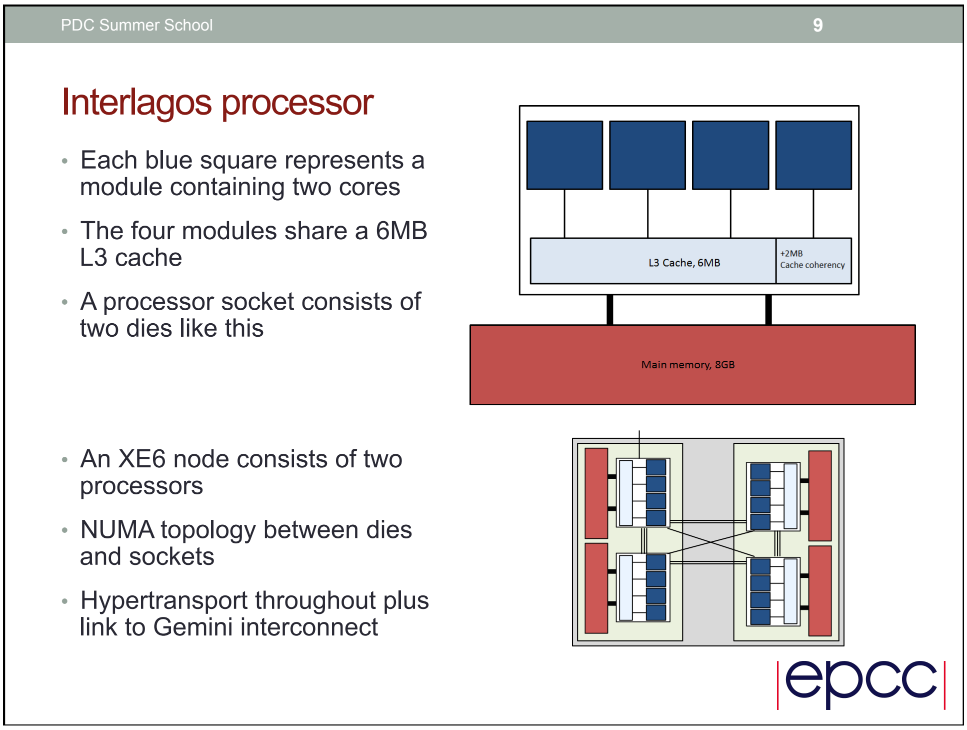
L1 16KB, 4-way data cache (4 cycles)

L2 TLB 1024 entries

L2 2MB, 16-way data cache (18-20 cycles)

🟦 16x x86_64 general purpose integer registers

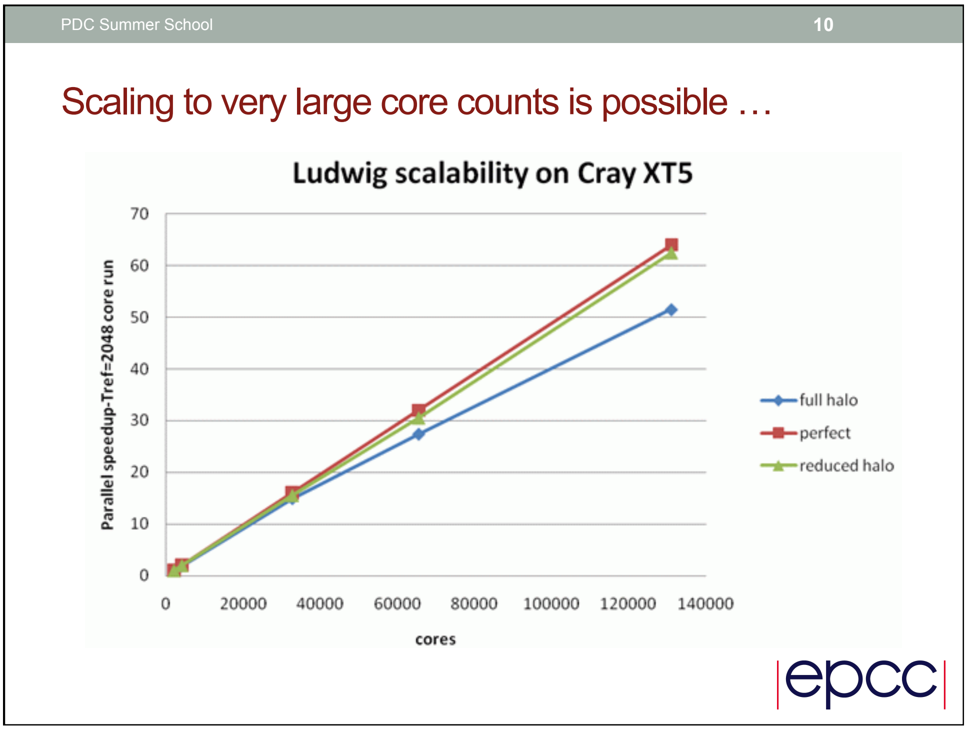
🟥 16x 256 bit AVX registers (ymm0-ymm15) / 16x 128 bit SSE registers (xmm0-xmm15)



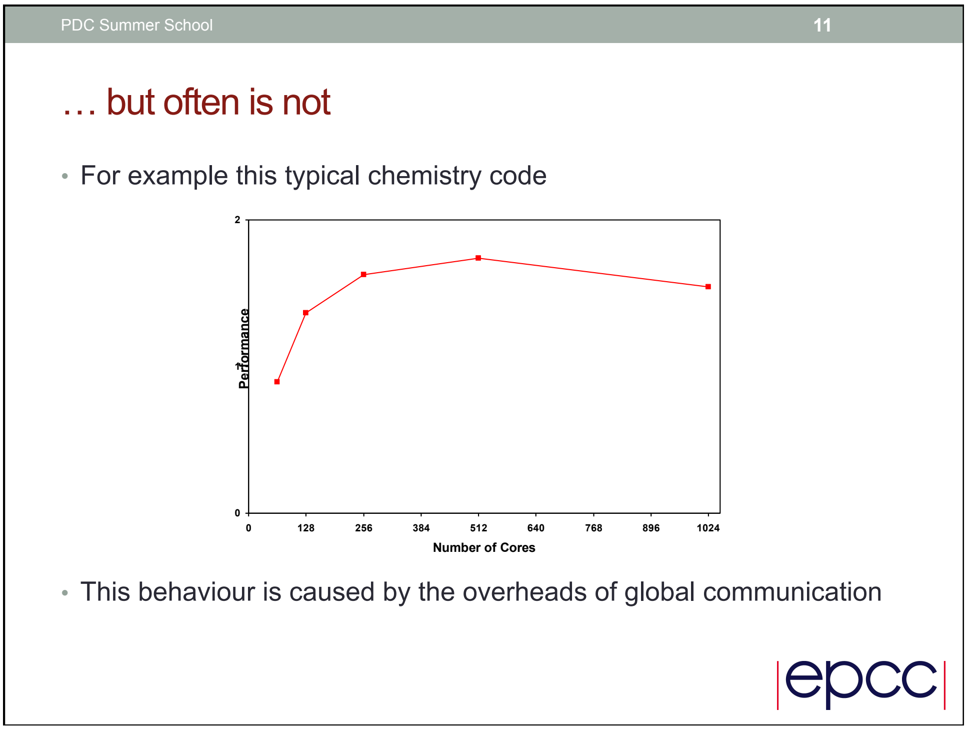
Interlagos processor

- Each blue square represents a module containing two cores
- The four modules share a 6MB L3 cache
- A processor socket consists of two dies like this

- An XE6 node consists of two processors
- NUMA topology between dies and sockets
- Hypertransport throughout plus link to Gemini interconnect



Scaling to very large core counts is possible …

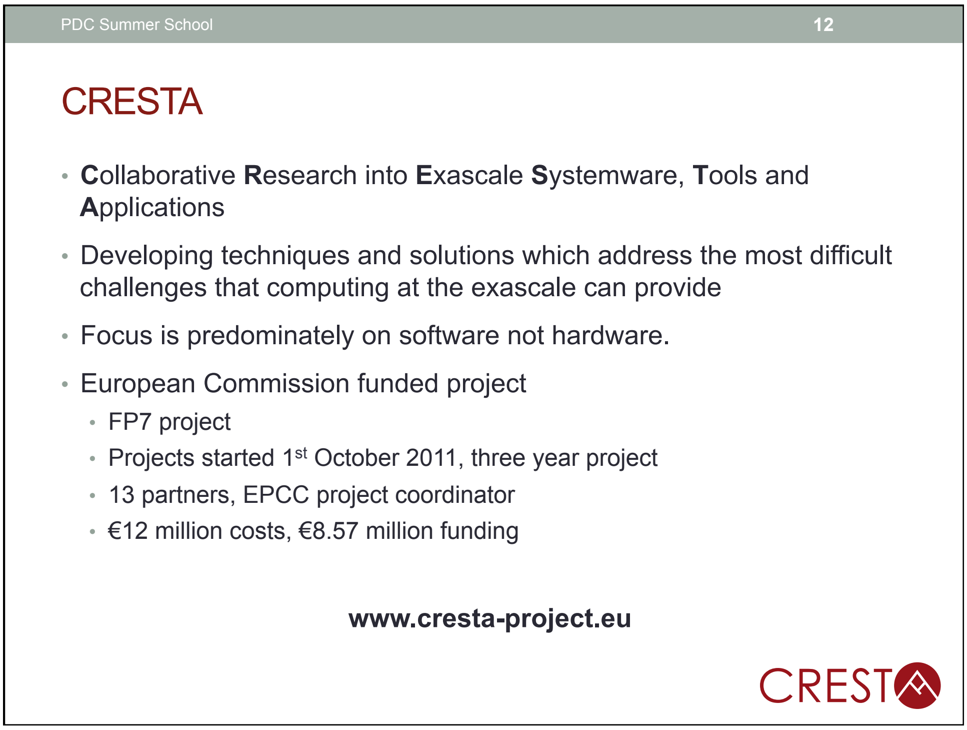


… but often is not

- For example this typical chemistry code

- This behaviour is caused by the overheads of global communication

epcc



CRESTA

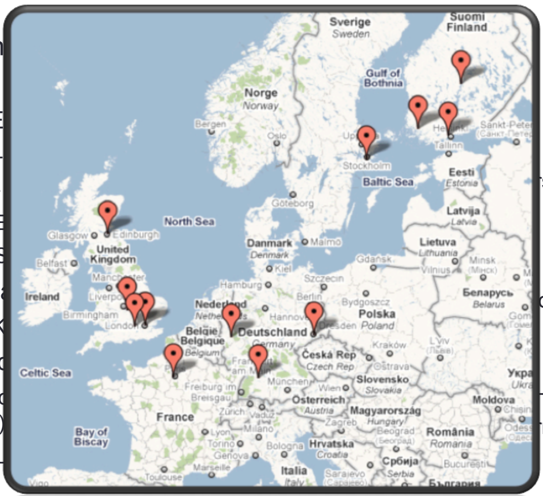
- **C**ollaborative **R**esearch into **E**xascale **S**ystemware, **T**ools and **A**pplications
- Developing techniques and solutions which address the most difficult challenges that computing at the exascale can provide
- Focus is predominately on software not hardware.
- European Commission funded 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 – … sity – Abo,
    - CSC – E … – Jyvaskyla,
    - PDC – S
  - A world le … ondon –
    - Cray UK … UK
  - World lead … – Paris, France
    - Technis … many
      (Vampir)
    - Allinea L

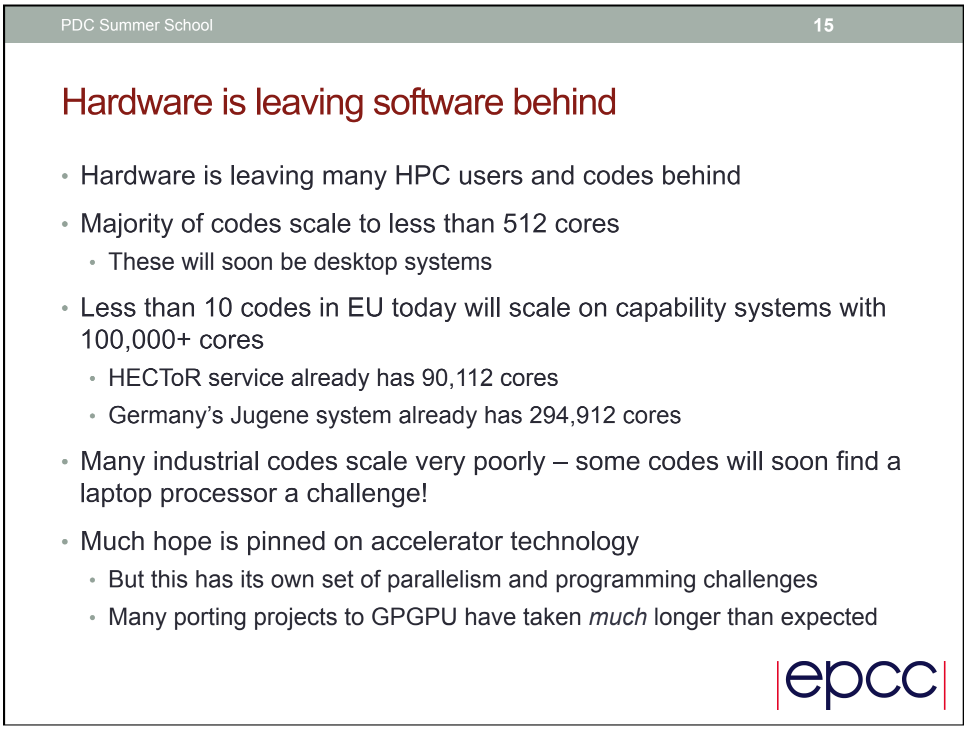


CRESTA

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

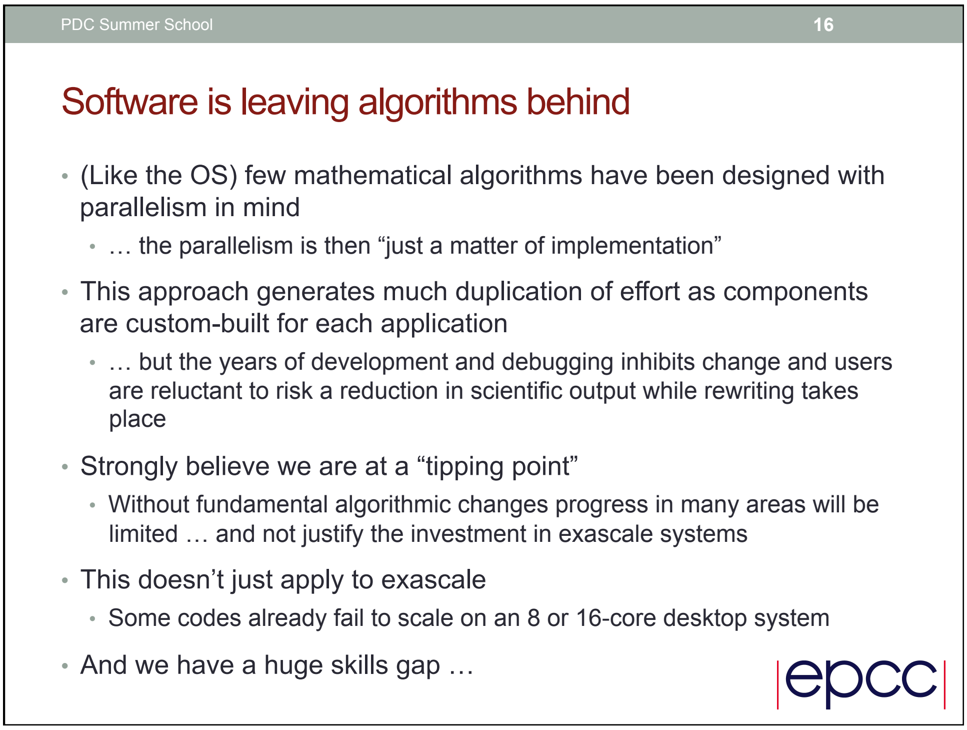
epcc



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

epcc



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 …

epcc

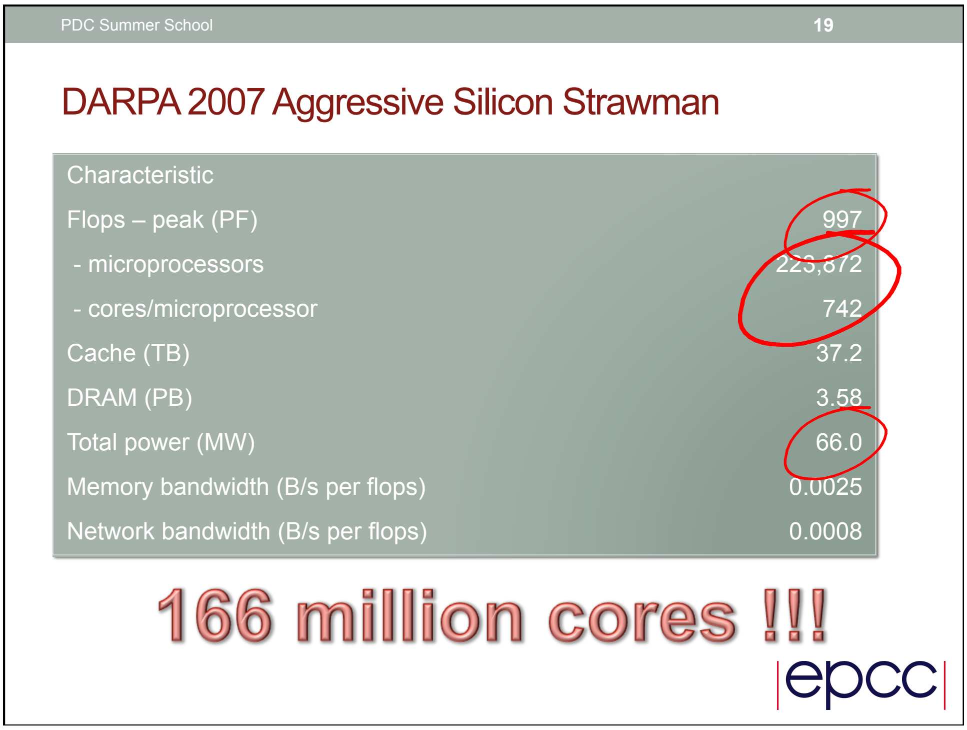
What are the challenges?

- 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

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

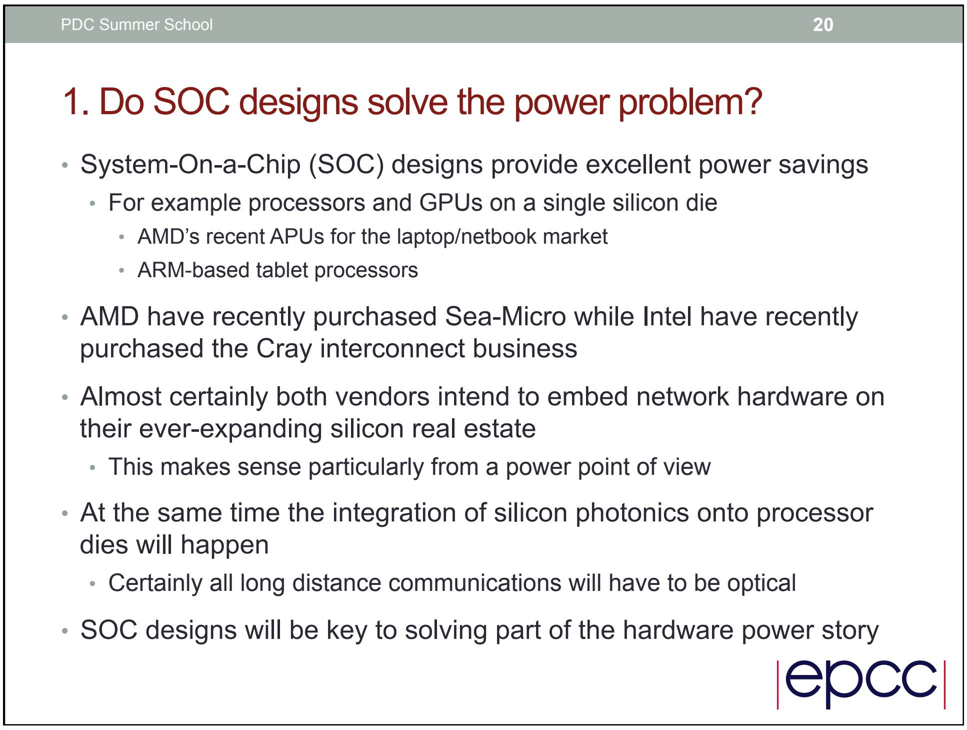


DARPA 2007 Aggressive Silicon Strawman

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 !!!

epcc



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

epcc

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

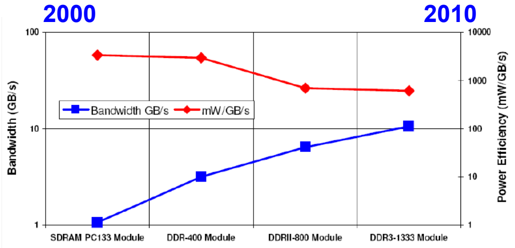
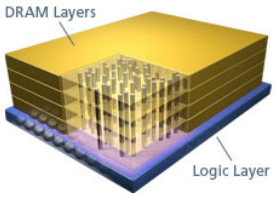


Figure 6.22: Commodity DRAM module power efficiency as a function of bandwidth.

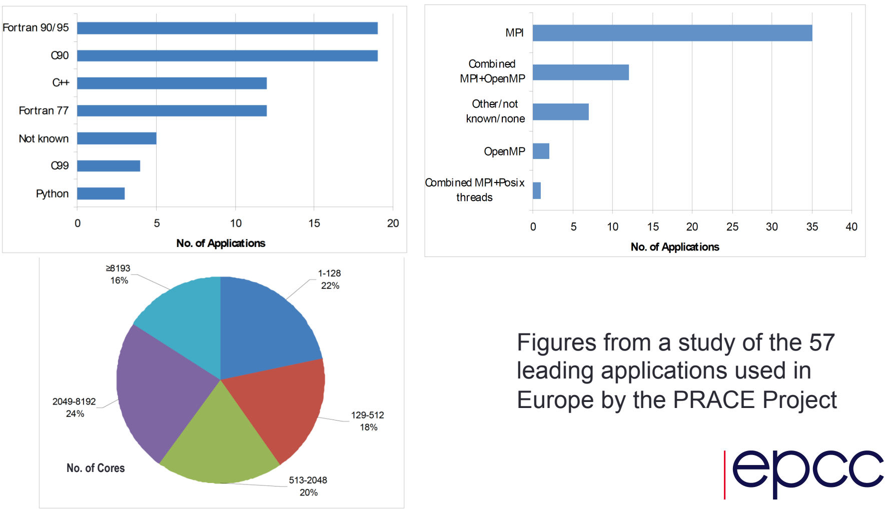
2: Memory performance

- 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



3. Application scalability

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



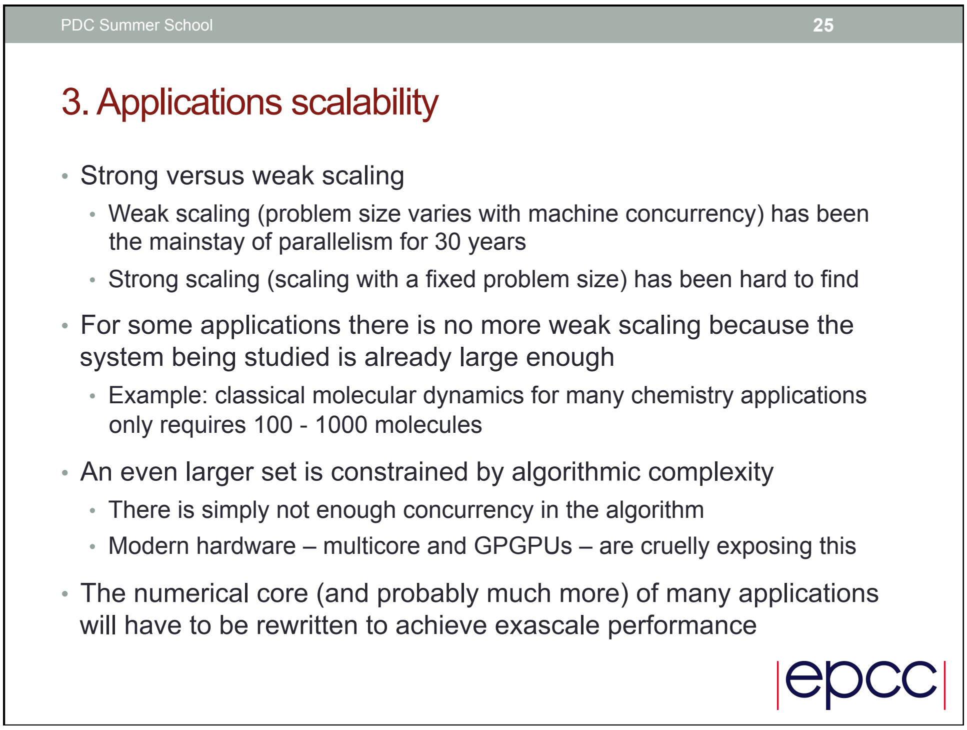
Figures from a study of the 57 leading applications used in Europe by the PRACE Project

epcc

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
  - New languages have been developed in USA but not in Europe
- 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

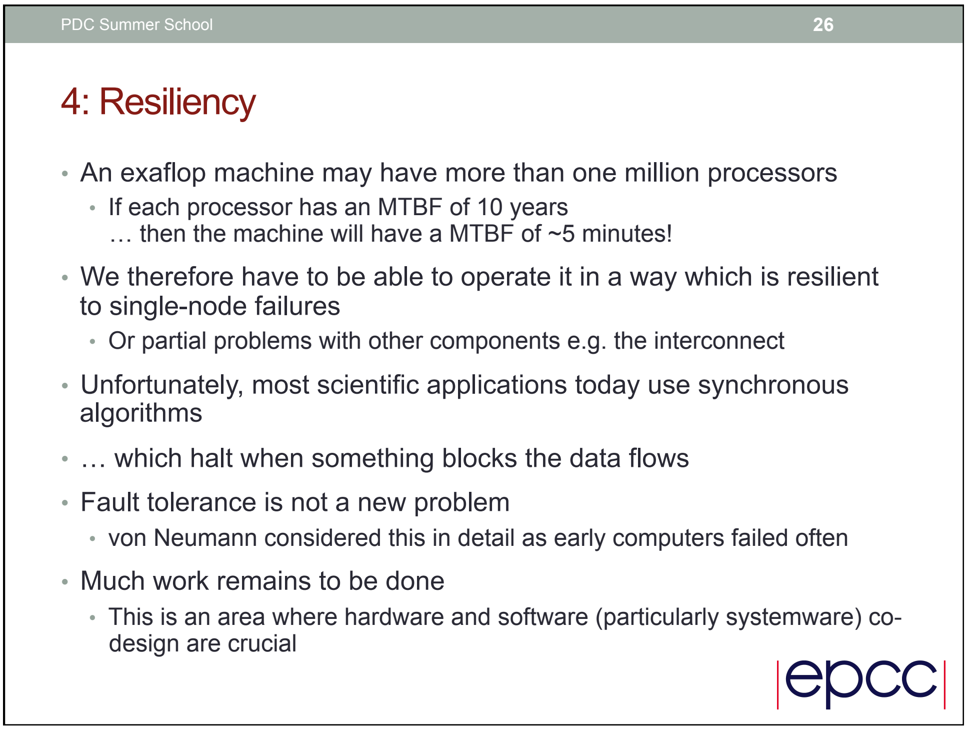
epcc



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 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

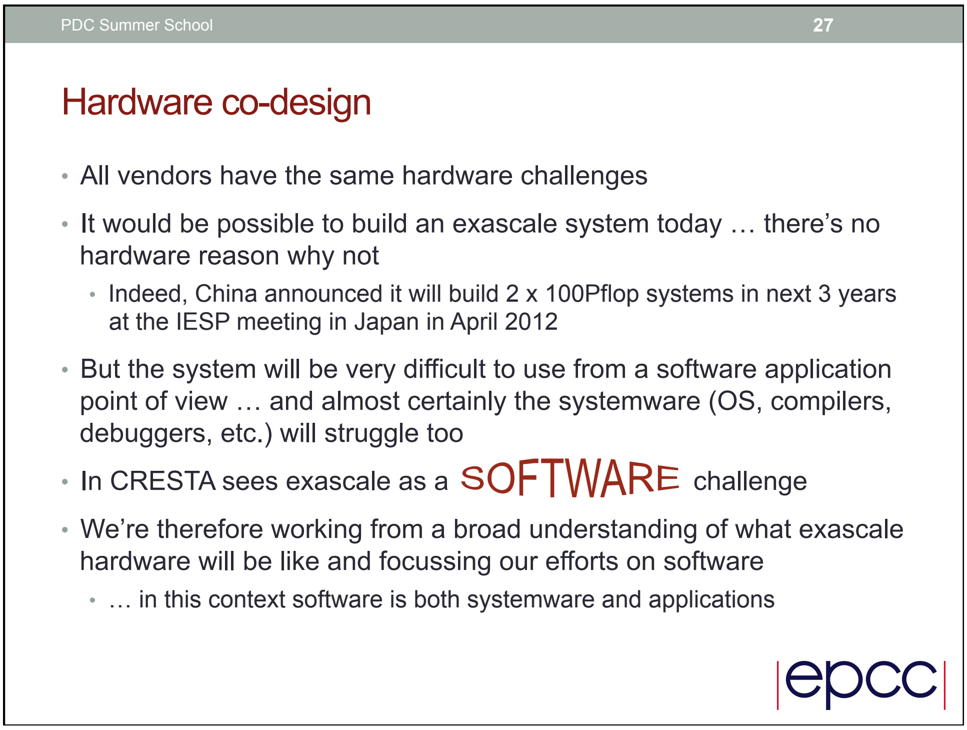
epcc



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

epcc

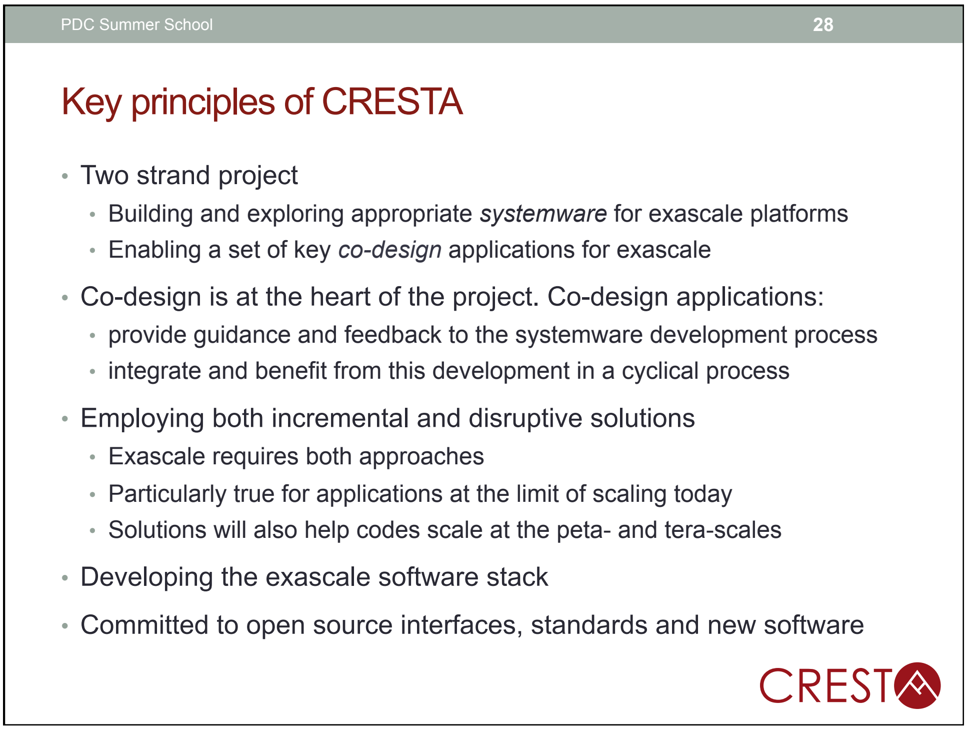


PDC Summer School 27

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

epcc

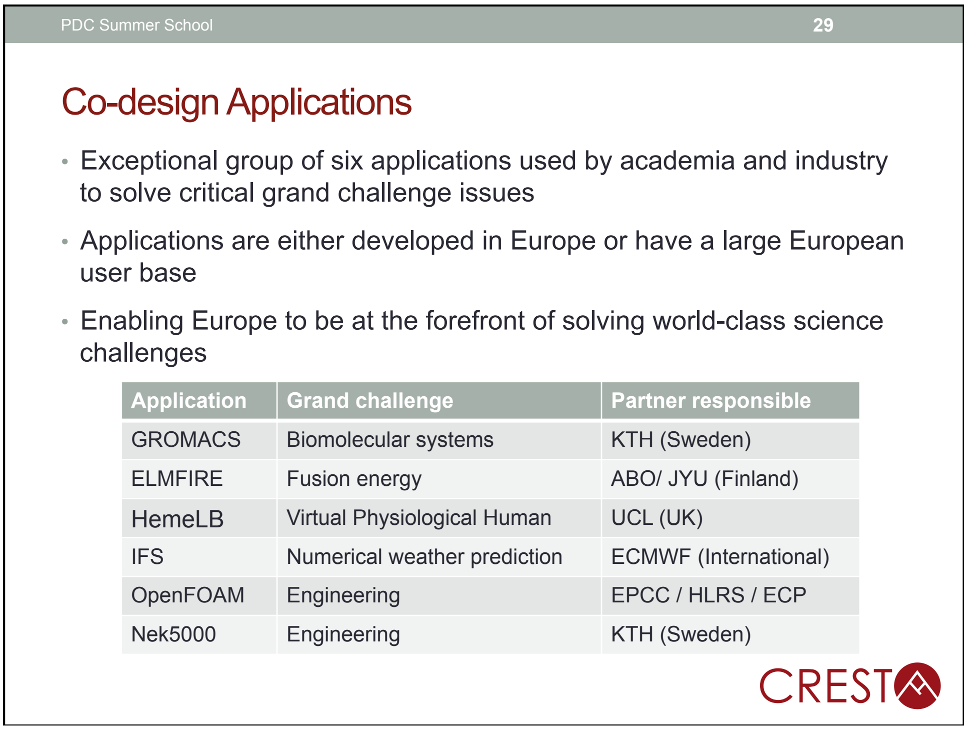


PDC Summer School 28

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

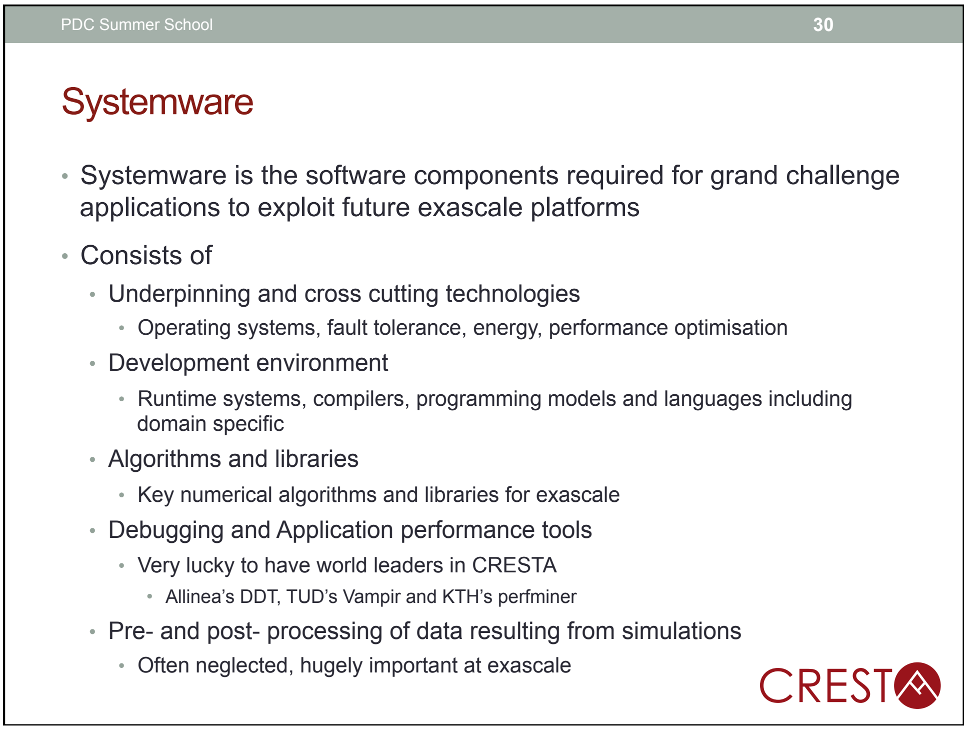
CRESTA



Co-design Applications

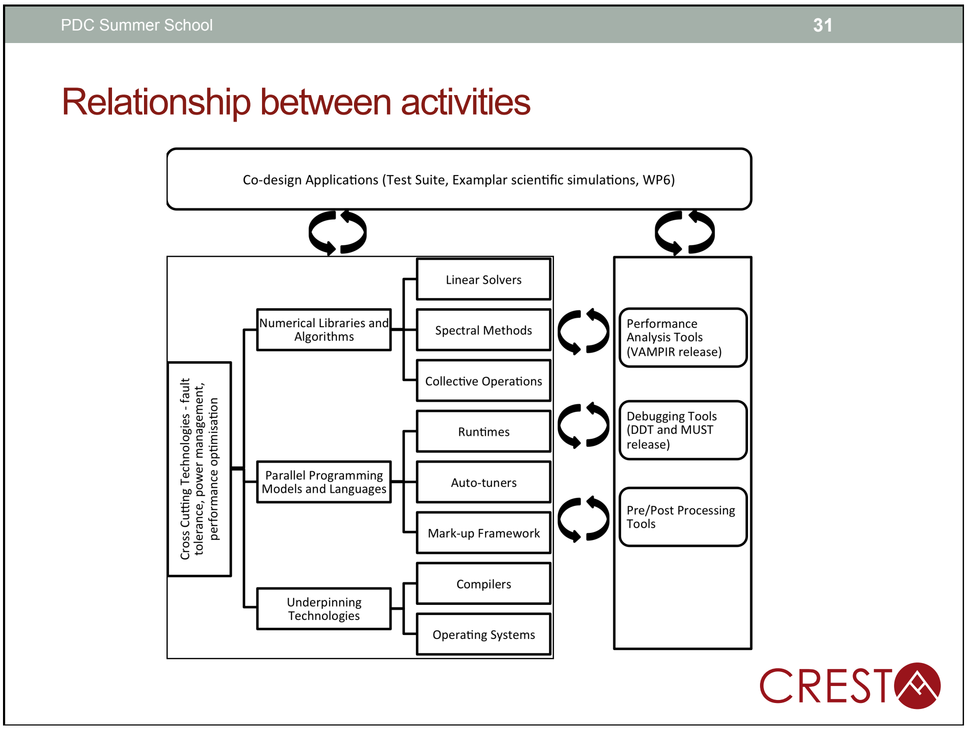
- Exceptional group of six applications used by academia and industry to solve critical grand challenge issues
- Applications are either developed in Europe or have a large European user base
- Enabling Europe to be at the forefront of solving world-class science challenges

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)



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

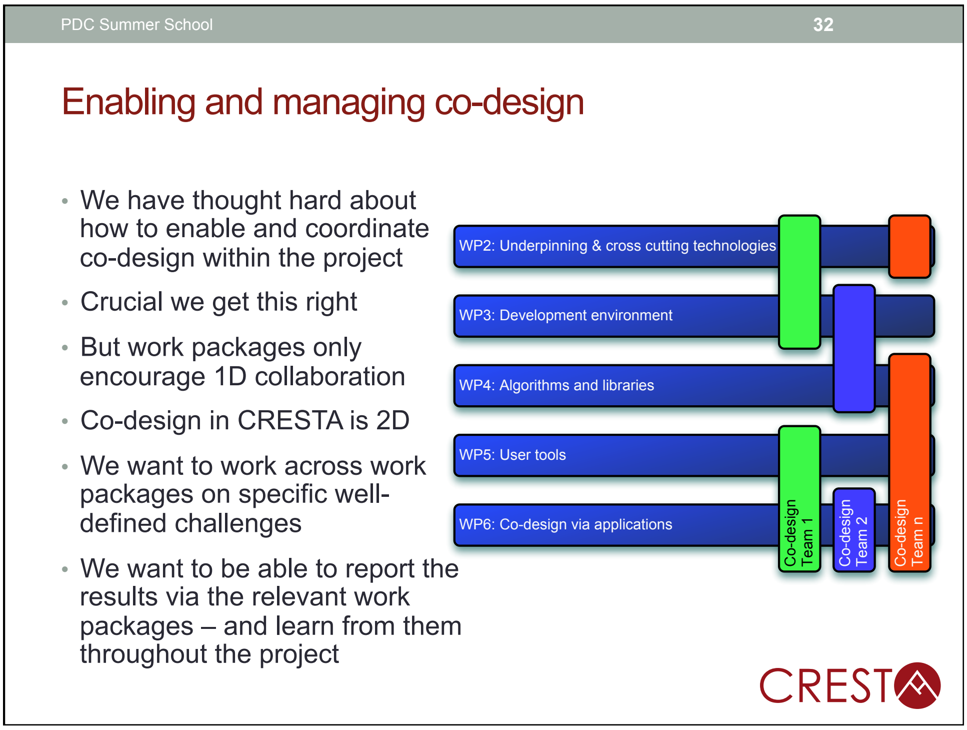


PDC Summer School 31

Relationship between activities

Co-design Applications (Test Suite, Examplar scientific simulations, WP6)
Cross Cutting Technologies - fault tolerance, power management, performance optimisation
Numerical Libraries and Algorithms
Linear Solvers
Spectral Methods
Collective Operations
Parallel Programming Models and Languages
Runtimes
Auto-tuners
Mark-up Framework
Underpinning Technologies
Compilers
Operating Systems
Performance Analysis Tools (VAMPIR release)
Debugging Tools (DDT and MUST release)
Pre/Post Processing Tools

CRESTA



PDC Summer School 32

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

WP2: Underpinning & cross cutting technologies
WP3: Development environment
WP4: Algorithms and libraries
WP5: User tools
WP6: Co-design via applications
Co-design Team 1
Co-design Team 2
Co-design Team n

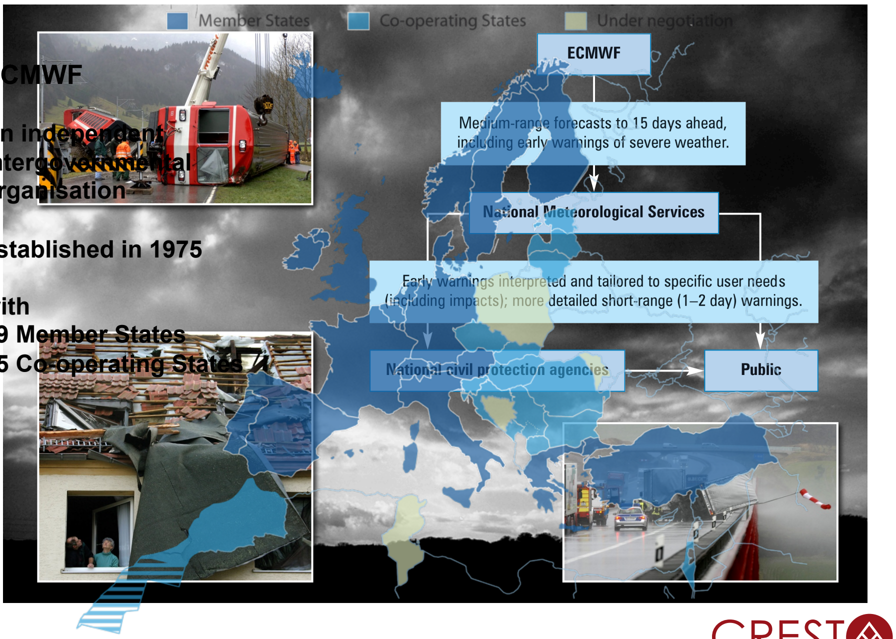
CRESTA

Example of incremental and disruptive approaches

- FFTs are a challenge at exascale because
  - Very large number of HPC applications use them
  - Distributed memory parallel FFT is already a major performance issue today – we accept some FFTs will not scale further
- Two approaches:

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

CREST



ECMWF

Medium-range forecasts to 15 days ahead, including early warnings of severe weather.

National Meteorological Services

Early warnings interpreted and tailored to specific user needs (including impacts); more detailed short-range (1–2 day) warnings.

National civil protection agencies

Public

ECMWF

An independent intergovernmental organisation

established in 1975

with
19 Member States
15 Co-operating States

CREST

IFS model: current and planned model resolutions

IFS model resolution	Envisaged Operational Implementation	Grid point spacing (km)	Time-step (seconds)	Estimated number of cores[1]
T1279 H[2]	2010 (L91) 2012 (L137)	16	600	1100 1600
T2047 H	2014-2015	10	450	6K
T3999 NH[3]	2020-2021	5	240	80K
T7999 NH	2025-2026	2.5	120	1M

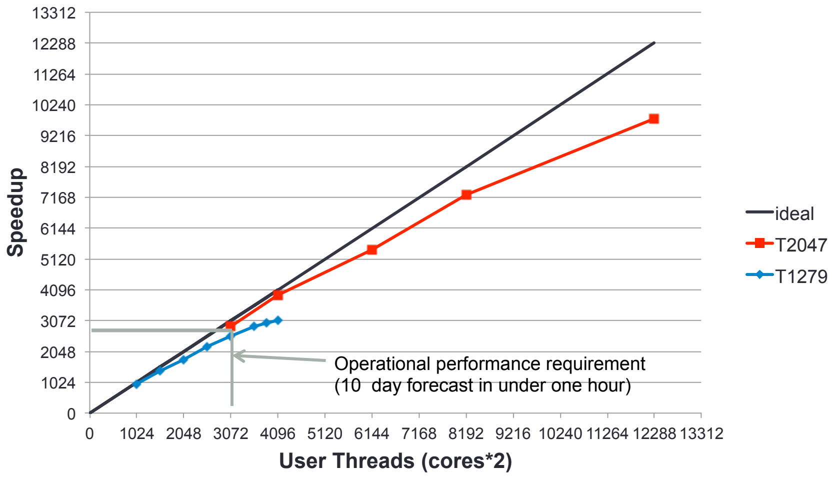
1 - a gross estimate for the number of 'Power7' equivalent cores needed to achieve a 10 day model forecast in under 1 hour (~240 FD/D), system size would normally be 10 times this number.

2 – Hydrostatic Dynamics

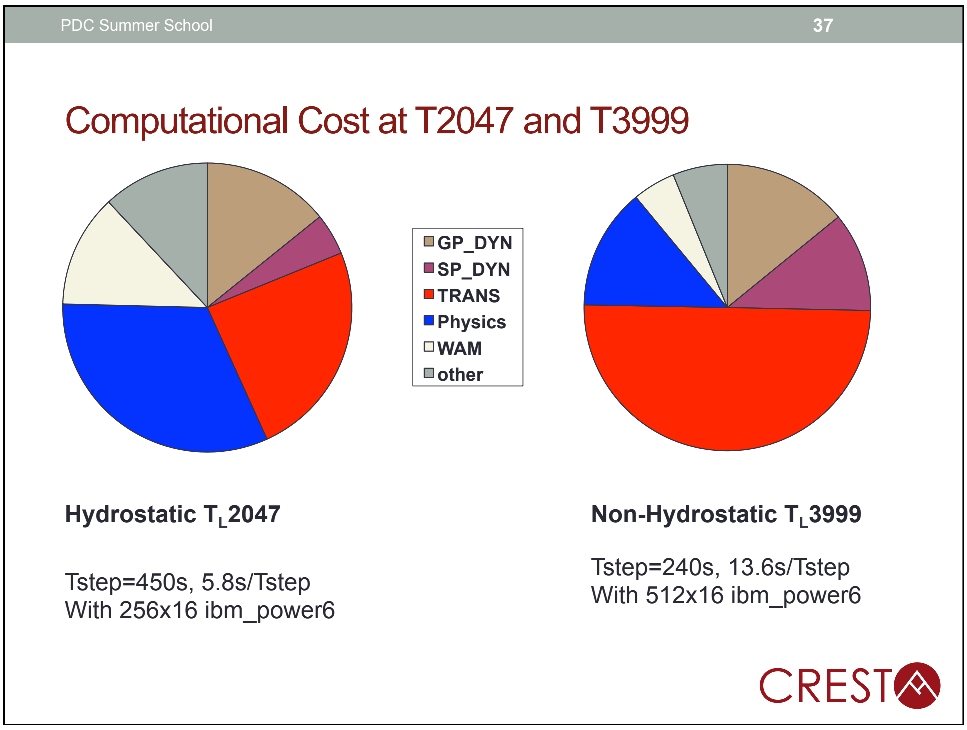
3 – Non-Hydrostatic Dynamics

CREST

IFS model speedup on IBM Power6 (~2010)



CREST



PDC Summer School 37

Computational Cost at T2047 and T3999

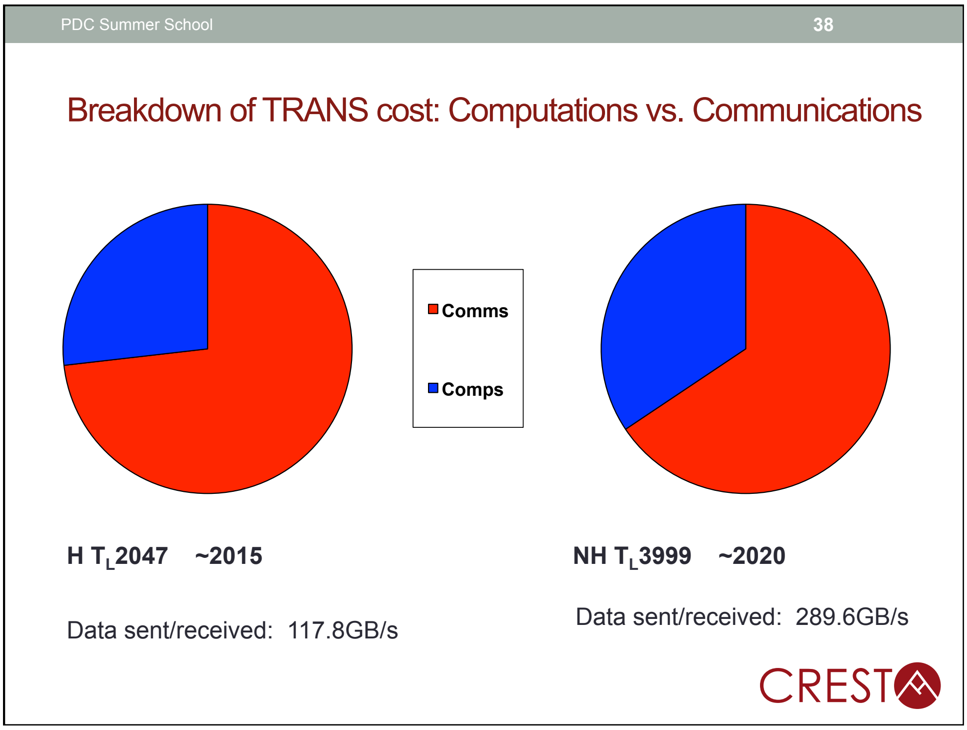
Hydrostatic T_L2047

Tstep=450s, 5.8s/Tstep
With 256x16 ibm_power6

Non-Hydrostatic T_L3999

Tstep=240s, 13.6s/Tstep
With 512x16 ibm_power6

CREST



PDC Summer School 38

Breakdown of TRANS cost: Computations vs. Communications

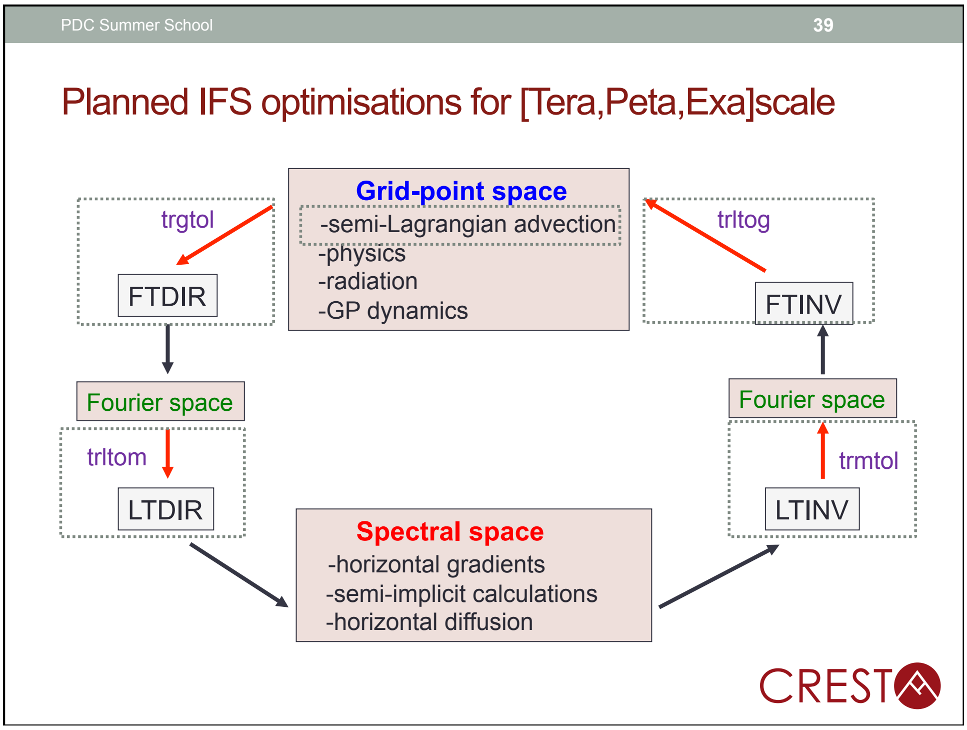
H T_L2047 ~2015

Data sent/received: 117.8GB/s

NH T_L3999 ~2020

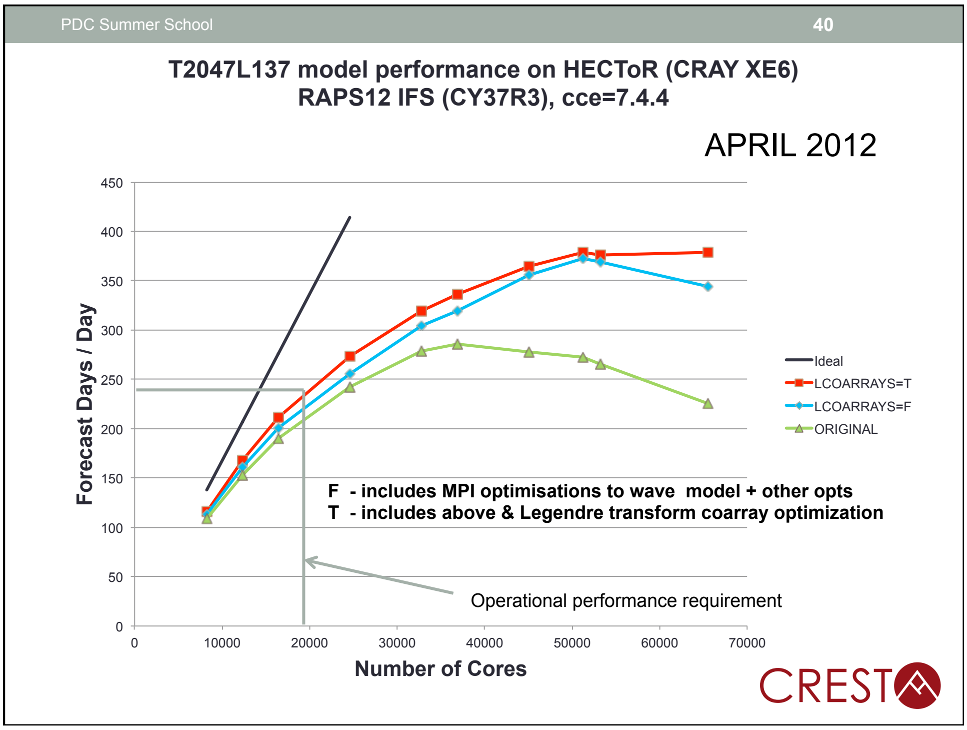
Data sent/received: 289.6GB/s

CREST



PDC Summer School 39

Planned IFS optimisations for [Tera,Peta,Exa]scale



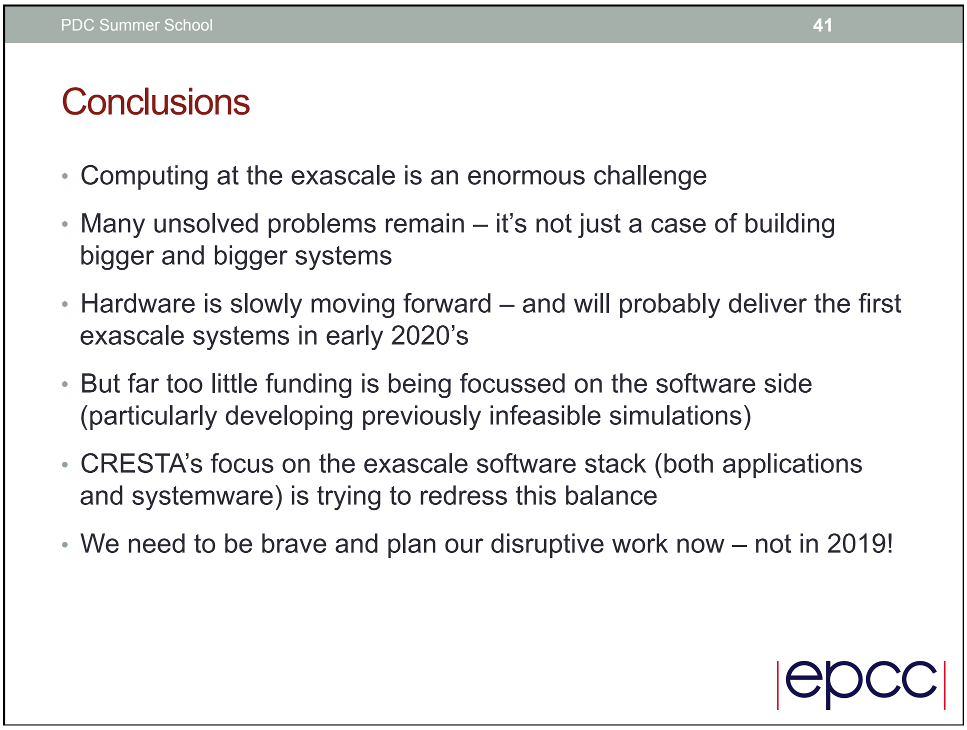
PDC Summer School 40

T2047L137 model performance on HECToR (CRAY XE6) RAPS12 IFS (CY37R3), cce=7.4.4

APRIL 2012

F - includes MPI optimisations to wave model + other opts
T - includes above & Legendre transform coarray optimization

Operational performance requirement



Conclusions

- Computing at the exascale is an enormous challenge
- Many unsolved problems remain – it's not just a case of building bigger and bigger systems
- Hardware is slowly moving forward – and will probably deliver the first exascale systems in early 2020's
- But 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!