Durham University
Department of Computer Science

The data movement challenge and a communication-avoiding mindset

IDAS Launch

T. Weinzierl

Agenda

Durham University
Department of Computer Science

. . . what scientific computing is

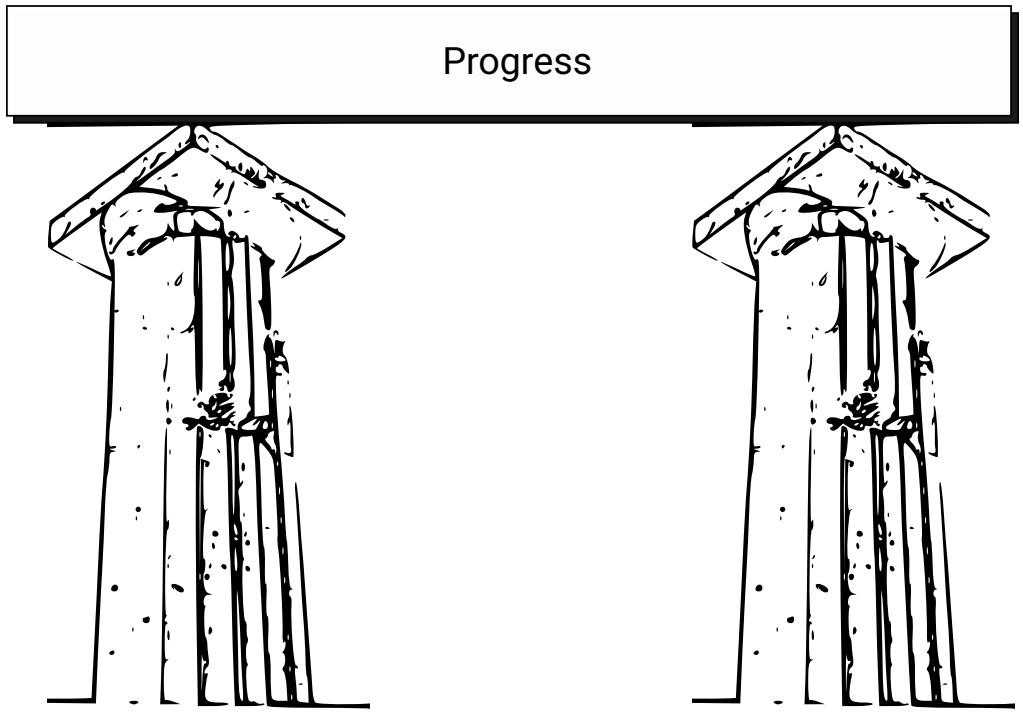
. . . which challenges hide around the corner

. . . why I think we are well-prepared (or not)

Leaning Tower

The virtualisation of science

Durham University
Department of Computer Science

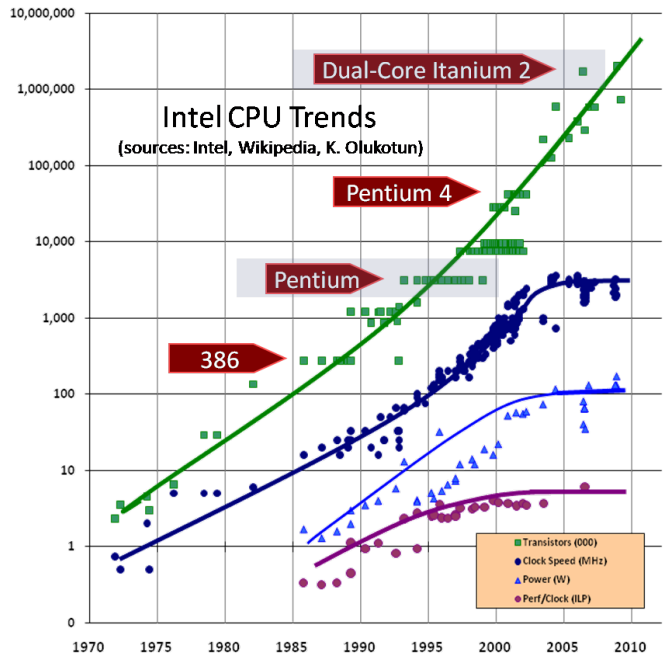


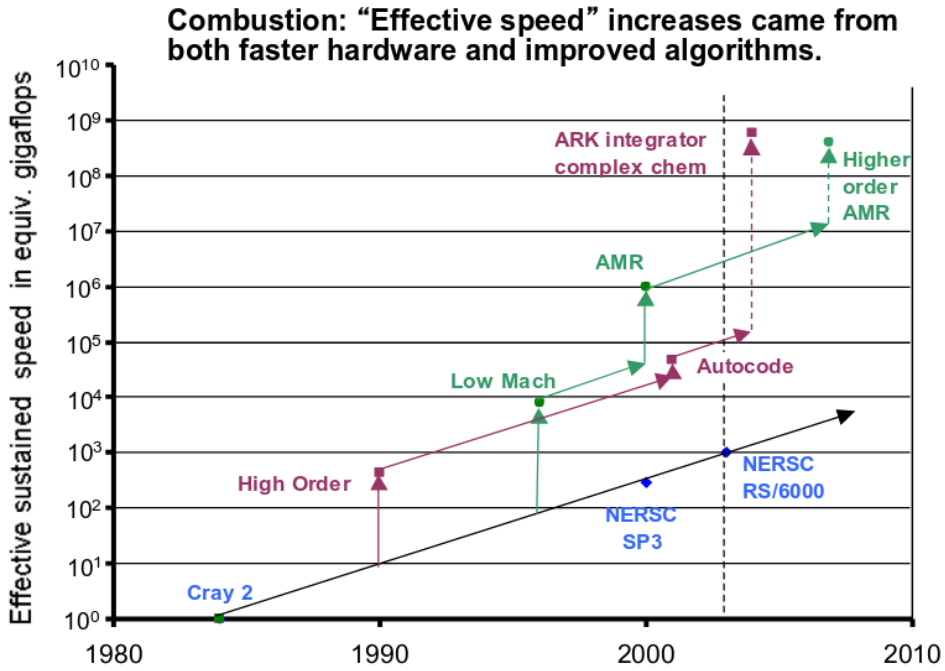
C. Johnson (SCI): Before the great discovery was the creation of a new tool!

(from the movie “The Golden Age of Computing”)

The two facets of “better tools”

Durham University
Department of Computer Science



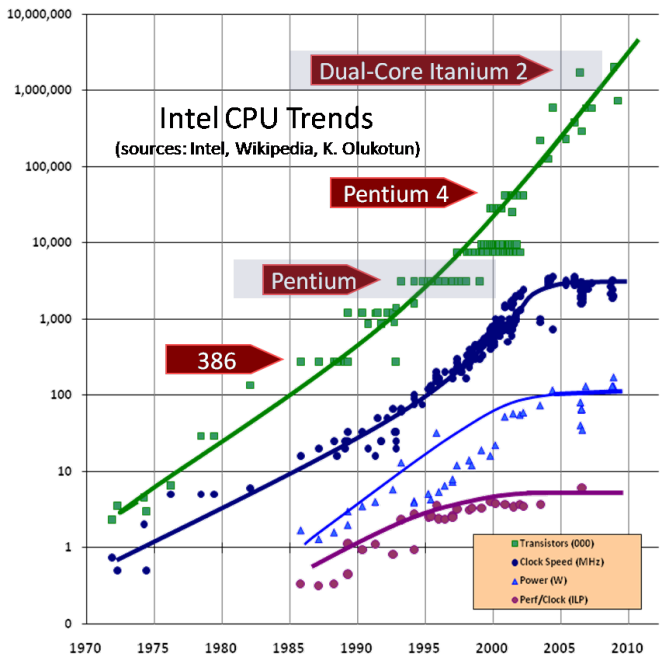


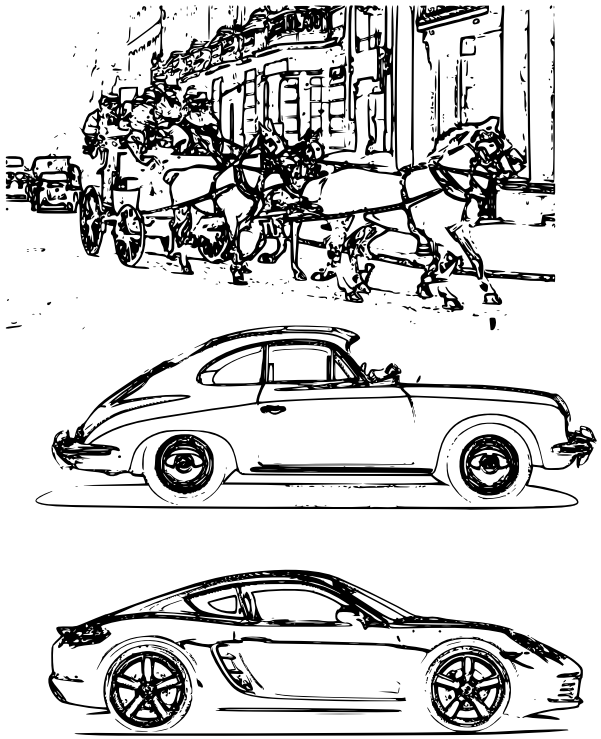
H. Sutter: The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software, 2005 (left)

D. Keyes: SCaLeS Report, Vol. 2, 2004 (right)

The Good, . . .

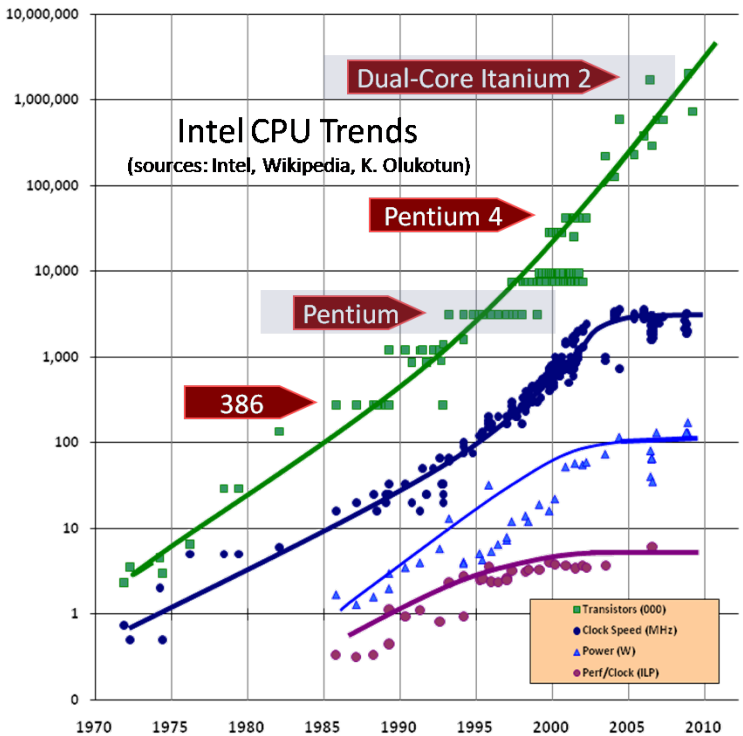
Durham University
Department of Computer Science

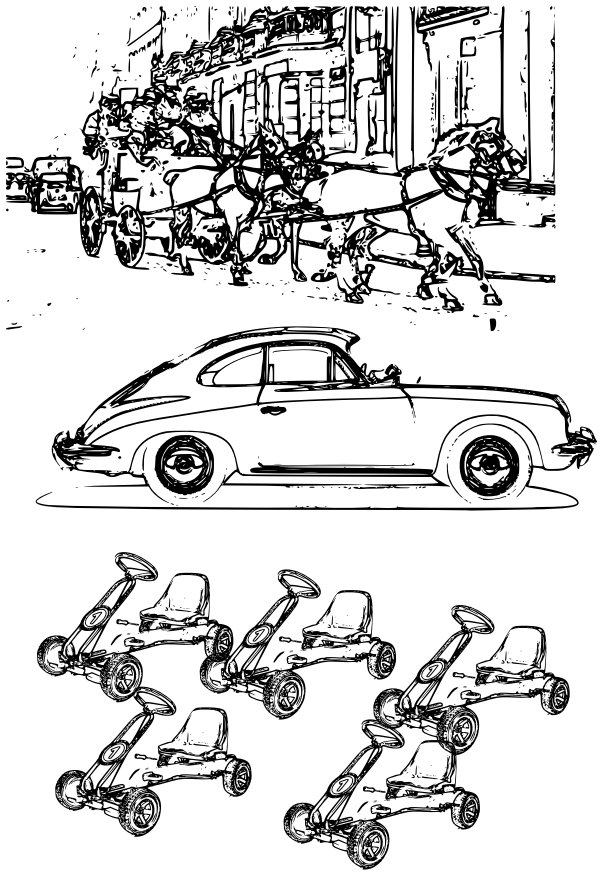




The Good, . . .

Durham University
Department of Computer Science

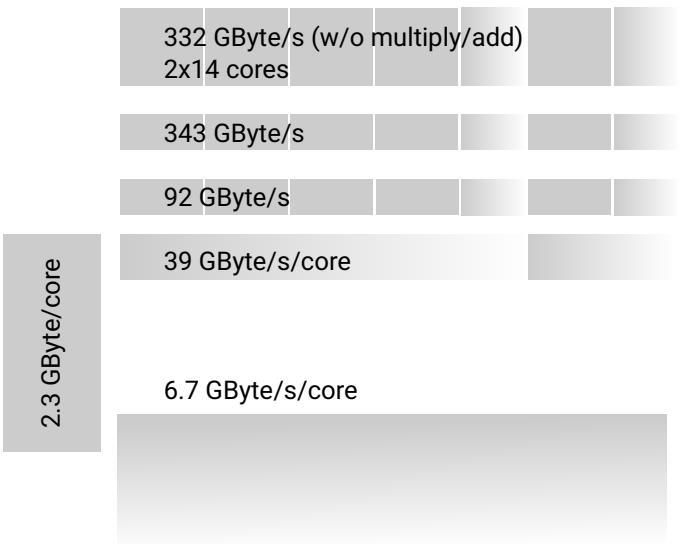




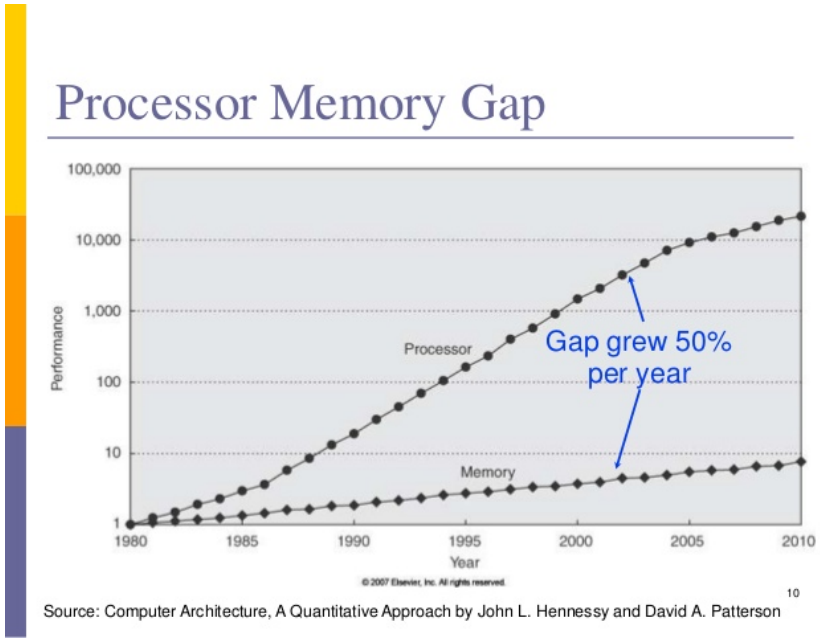
We had to rewrite all of our software!

The Good, the Bad, . . .

Durham University
Department of Computer Science



SuperMUC phase 2

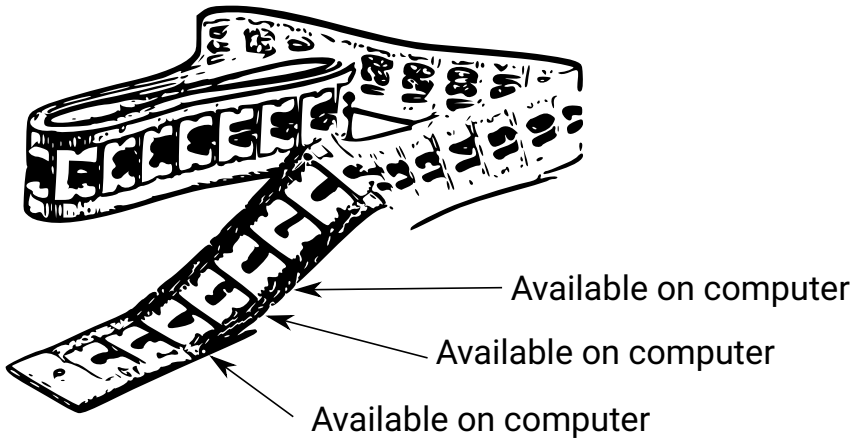


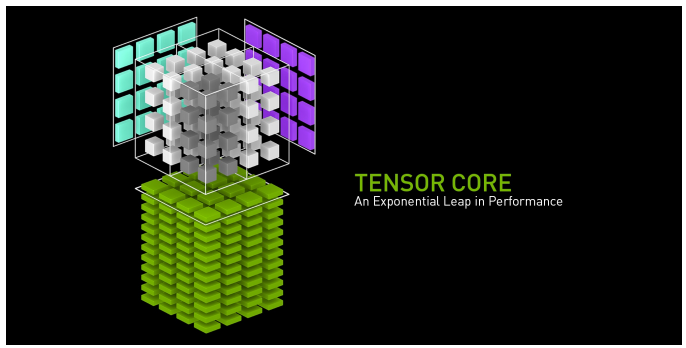
Source: Computer Architecture, A Quantitative Approach by John L. Hennessy and David A. Patterson

We will have to rewrite all of our software!

The Good, the Bad and the Ugly

Durham University
Department of Computer Science





We don’t know yet how to rewrite all of our software!

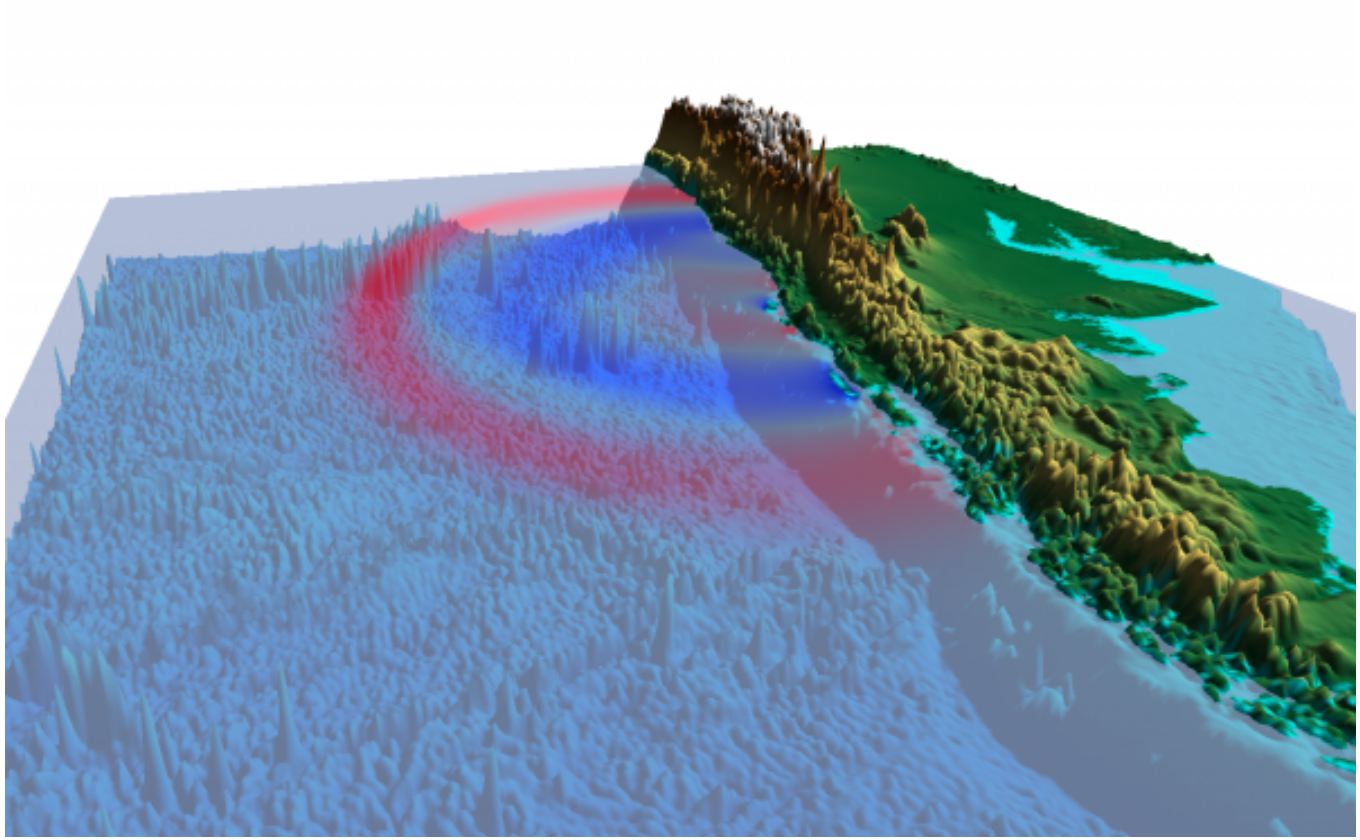
Vision: Allow groups with decent computational background to write an exascale solver for

$$\boldsymbol{M}\frac{\partial}{\partial t}Q + \boldsymbol{\nabla}\cdot\mathbf{F}(Q) + \sum_i \boldsymbol{\mathcal{B}}_i \frac{\partial Q}{\partial x_i} = \mathbf{S} + \sum \delta$$

within a year.

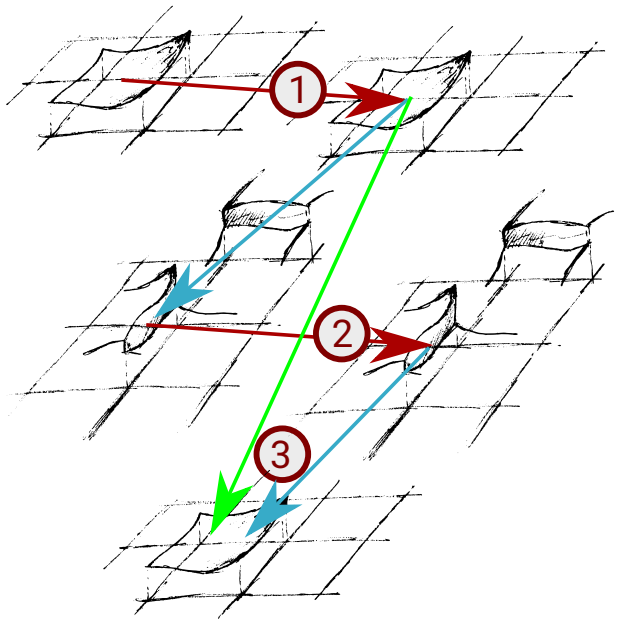
www.youtube.com: hpcsoftware

Durham University
Department of Computer Science



The ADER-DG algorithm pattern

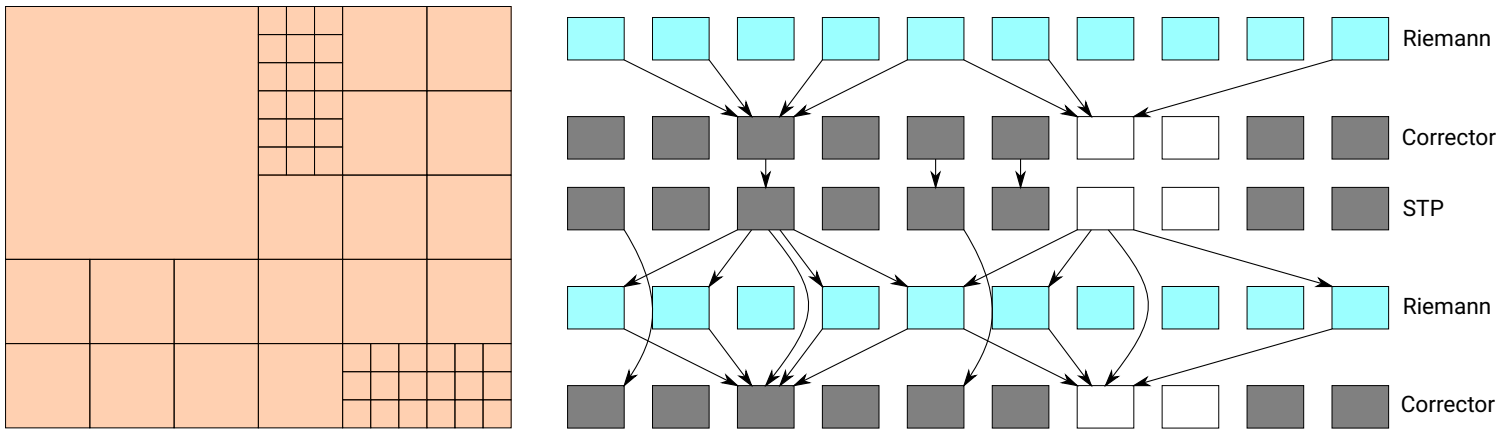
Durham University
Department of Computer Science



1. Predictor Weak space-time ansatz
   - span full space-time polynomial (locally)
   - yields/amplifies jumps in solution
2. Riemann
   - Riemann on interfaces
3. Corrector
   - Sophisticated sum of two results
   - Includes CFL/admissibility checks

HPC's lingua franca: ADER-DG as task graph

Durham University
Department of Computer Science



Observations:

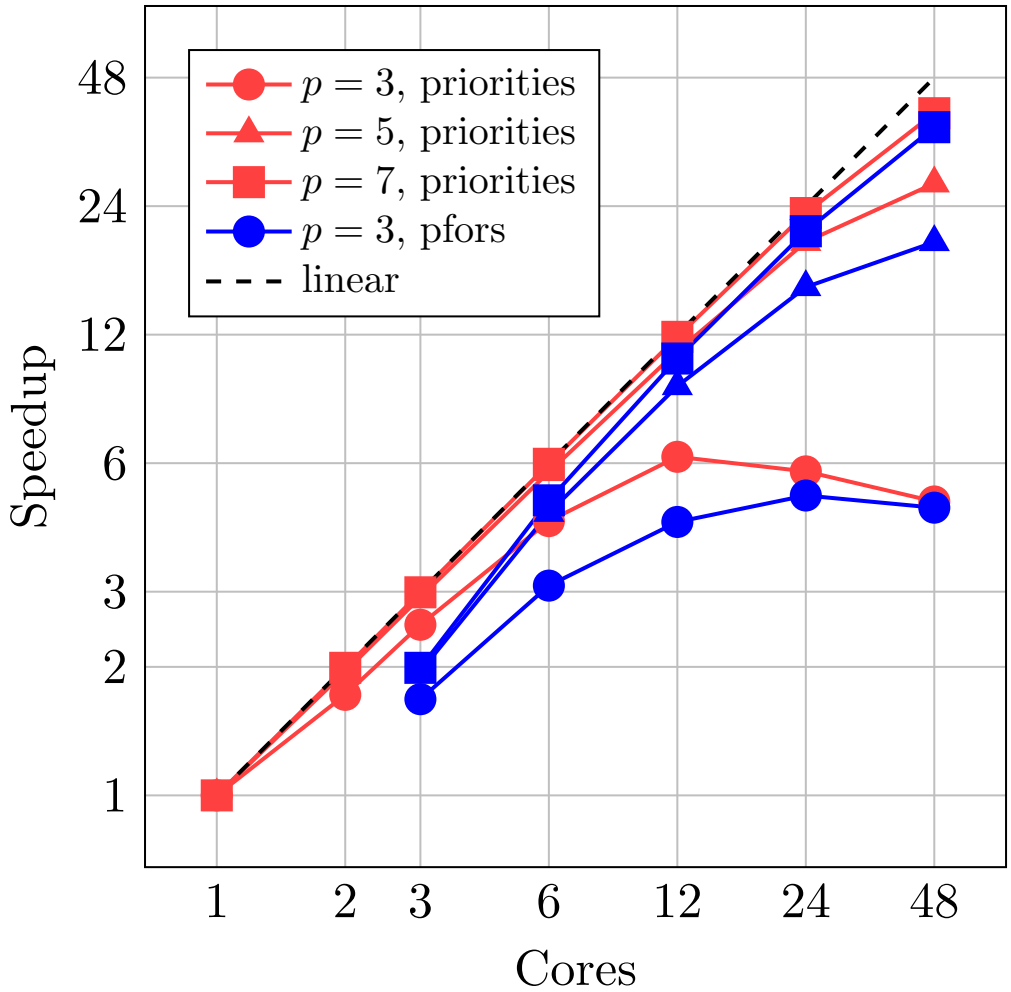
- ADER-DG describes a task pattern
- Mesh instantiates task graph from pattern

Actions:

- Identify tasks
- Characterise tasks
- Find independent tasks

A really fast wave equation solver

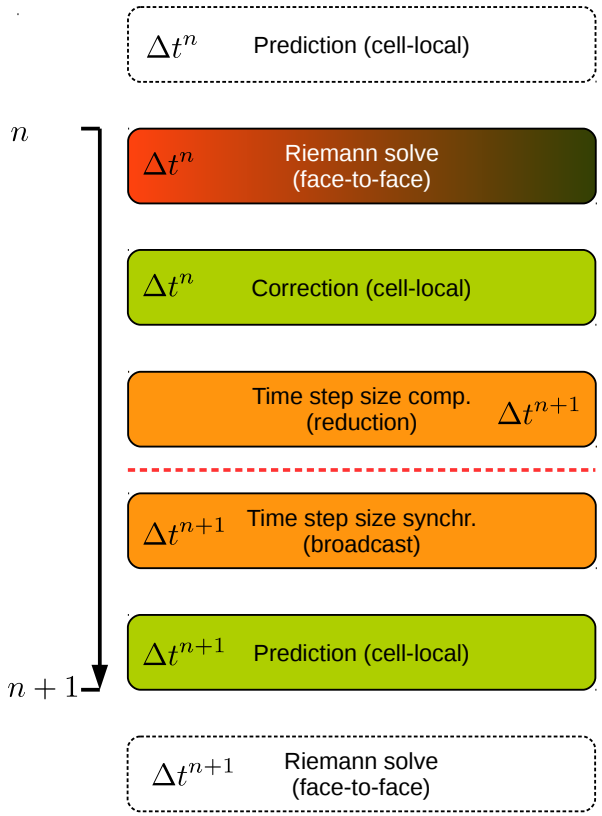
Durham University
Department of Computer Science



We have **not** written the fastest code. We have created a code combining a set of features that no other code provides (functional level). And this code is fast (non-functional).

Shifted tasking: be optimistic and anarchic

Durham University
Department of Computer Science



For the time being: assume Δt is known and there's no need to reduce it

Shifted algorithm:

- First read of face: trigger Riemann
- Enter cell: correct solution
  (all $2d$ faces are read already)
- Trigger subsequent STP immediately

Observations:

- Shifted execution model brings tasks optimistically forward

$\Rightarrow$ Eliminate synchronisation

- Only two (logical) steps (task fusion)

$\Rightarrow$ Increase AI

- Amortised single-touch semantics

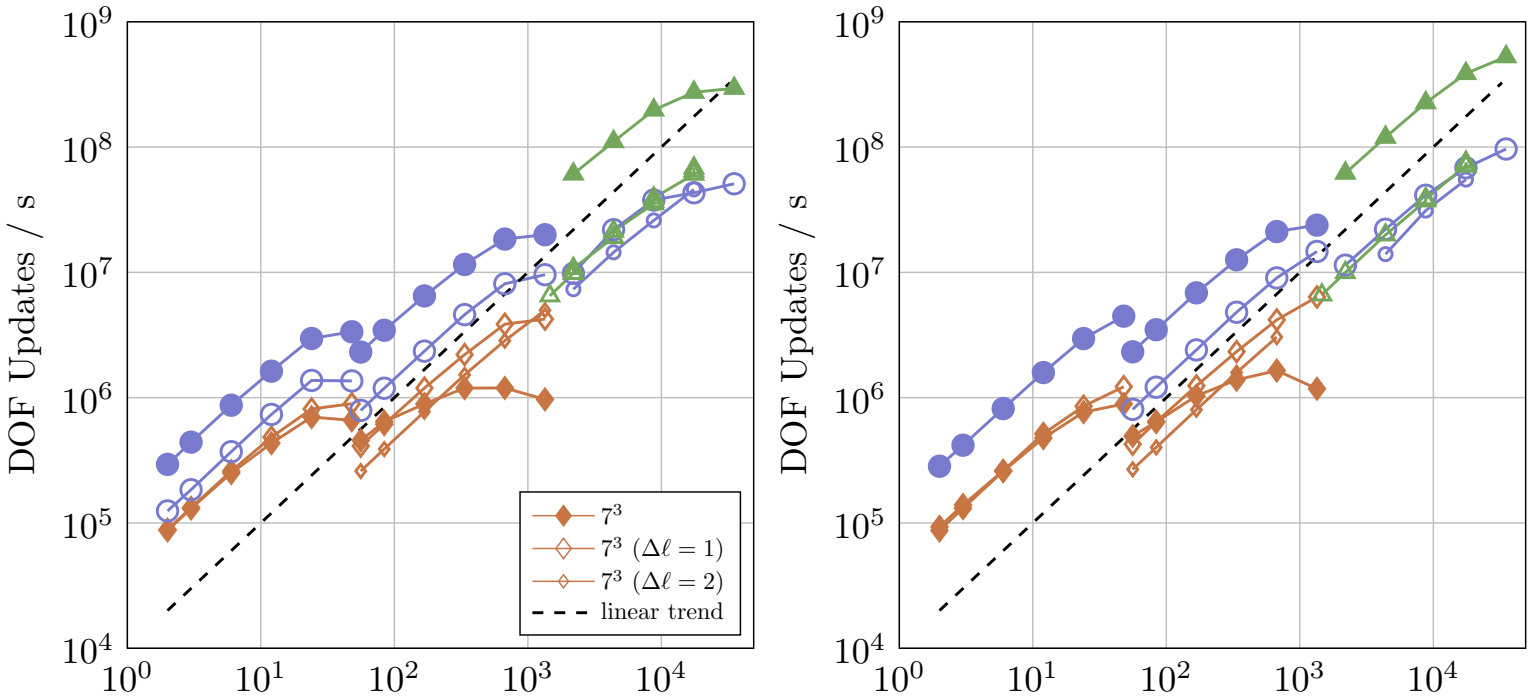
$\Rightarrow$ Eliminate multiple reads

- Parallel traversal intermixes steps

$\Rightarrow$ Memory access homogenisation

A really fast wave equation solver (again)

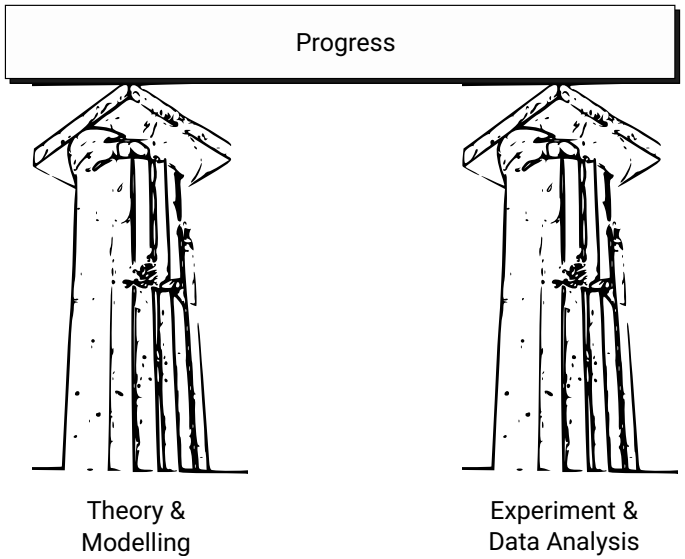
Durham University
Department of Computer Science



x-axis: number of cores; SuperMUC-NG (no. 2 in Europe)

The virtualisation of science continues

Durham University
Department of Computer Science



C. Johnson (SCI): Before the great discovery was the creation of a new tool!

(from the movie “The Golden Age of Computing”)

Sutter, Dongarra, many more: Our tools will fail us!

(prediction more than 15 years old)

The Unholy Three

Durham University
Department of Computer Science

	Scientific Computing	Machine Learning
Concurrency explosion	Omnipresent	Reason for success
Widening memory gap	CA techniques (work in progress)	Somehow ignored
Reduced precision	Open (unless in linear algebra)	Reason for success