ETH zürich

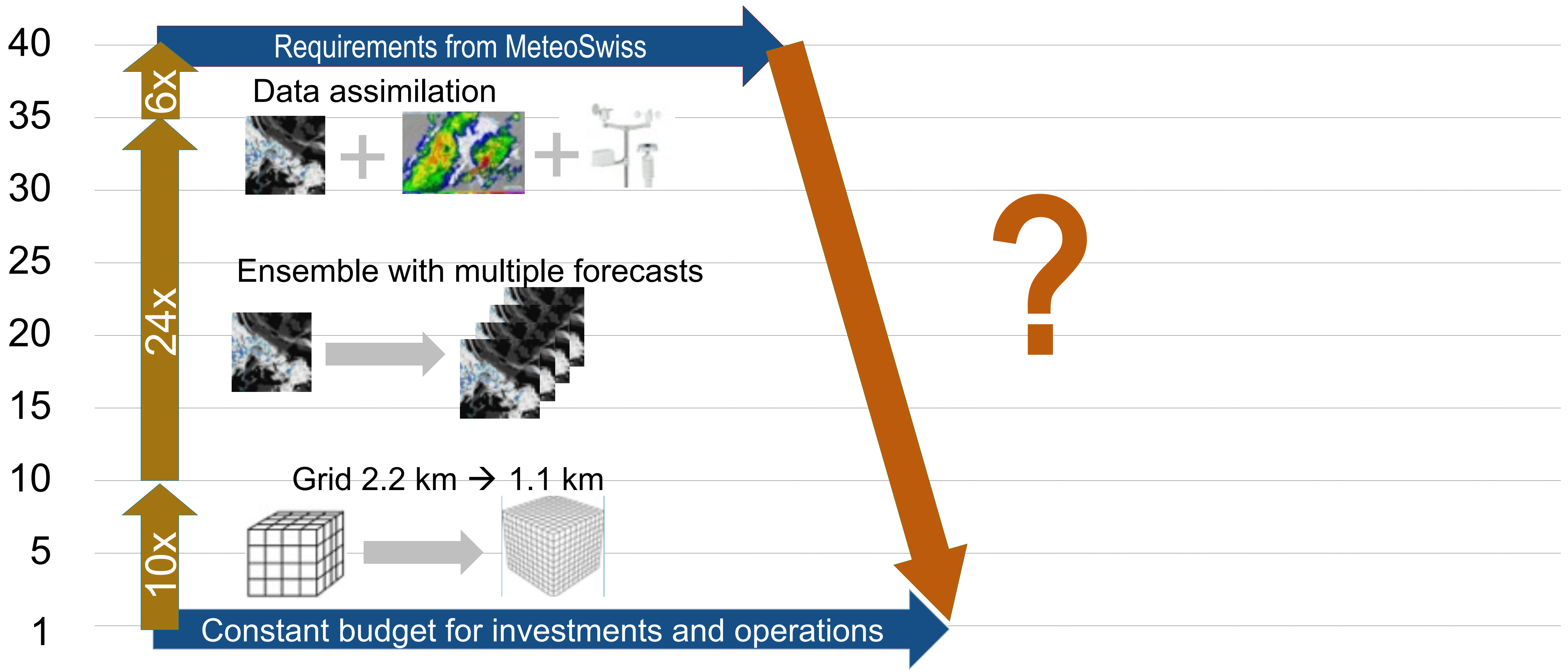
Exascale computing: endgame or new beginning for climate modelling

Thomas C. Schulthess

Operational system for NWP at CSCS

Albis & Lema (in production through 3/2016)   New system: Kesch & Escha

MeteoSwiss' performance ambitions



We need a 40x improvement between 2012 and 2015 at constant cost

ETH zürich

HPC wire

September 15, 2015

Today's Outlook: GPU-accelerated Weather Forecasting

John Russell

MeteoSwiss New Weather Supercomputer

World's First GPU-Accelerated Weather Forecasting System

2x Racks

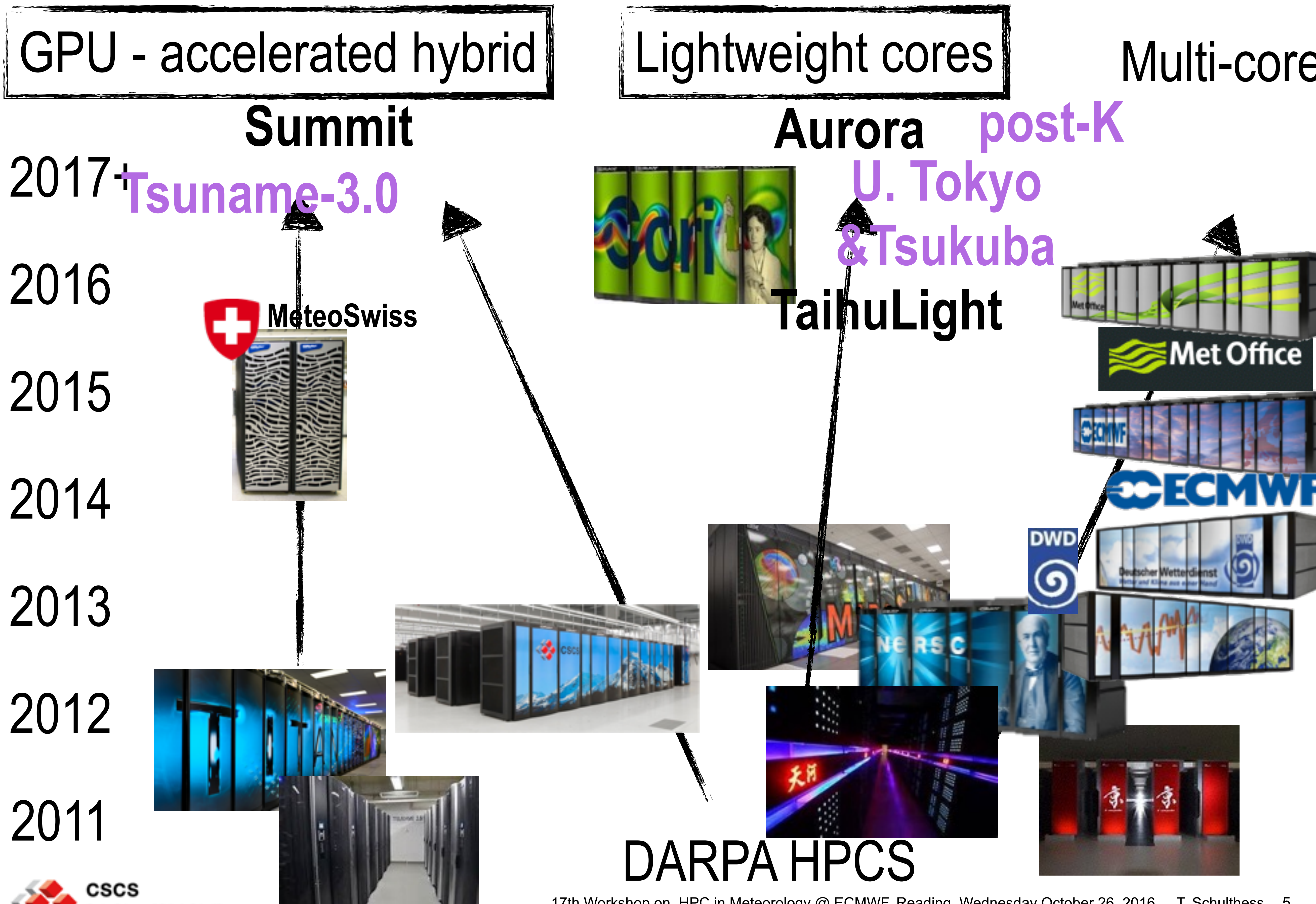
48 CPUs

192 Tesla K80 GPUs

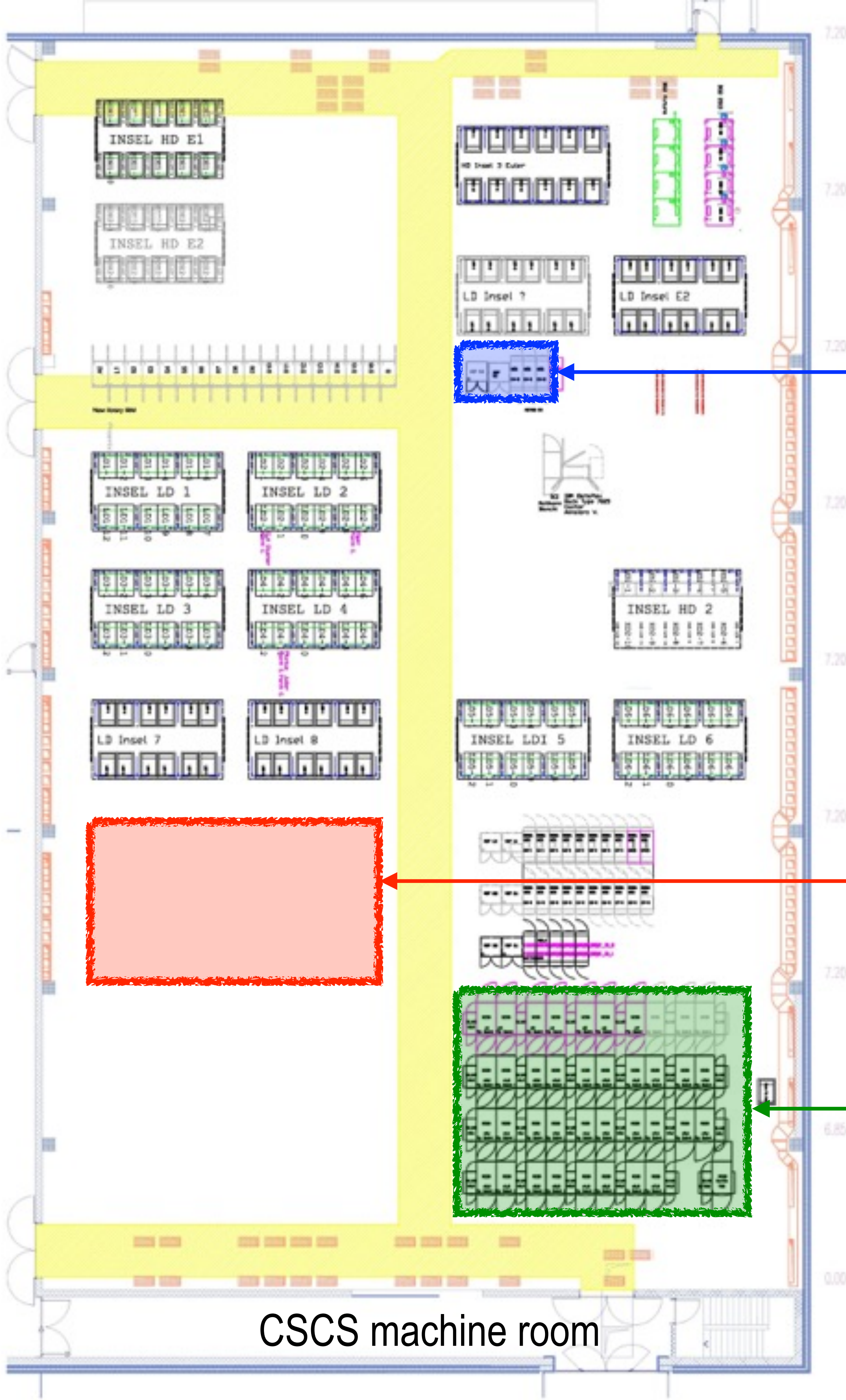
> 90% of FLOPS from GPUs

Operational in 2016

5 NVIDIA


GPU - accelerated hybrid
Lightweight cores
Multi-core
Summit
Aurora
post-K
U. Tokyo
&Tsukuba
2017+
Tsuname-3.0
2016
MeteoSwiss
TaihuLight
Met Office
2015
ECMWF
2014
DWD
2013
2012
2011
DARPA HPCS

State of the art implementation of new system for MeteoSwiss



CSCS machine room

Albis & Lema: 3 cabinets Cray XE6 installed Q2/2012

- New system needs to be installed Q3/2015
- Assuming 2x improvement in per-socket performance: ~20x more X86 sockets would require 30 Cray XC cabinets

New system for Meteo Swiss if we build it like the German Weather Service (DWD) did theirs, or UK Met Office, or ECMWF … (30 racks XC)

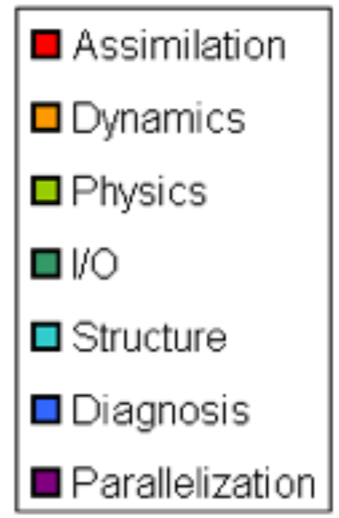
Current Cray XC30/XC40 platform (space for 40 racks XC)

Thinking inside the box was not a good option!

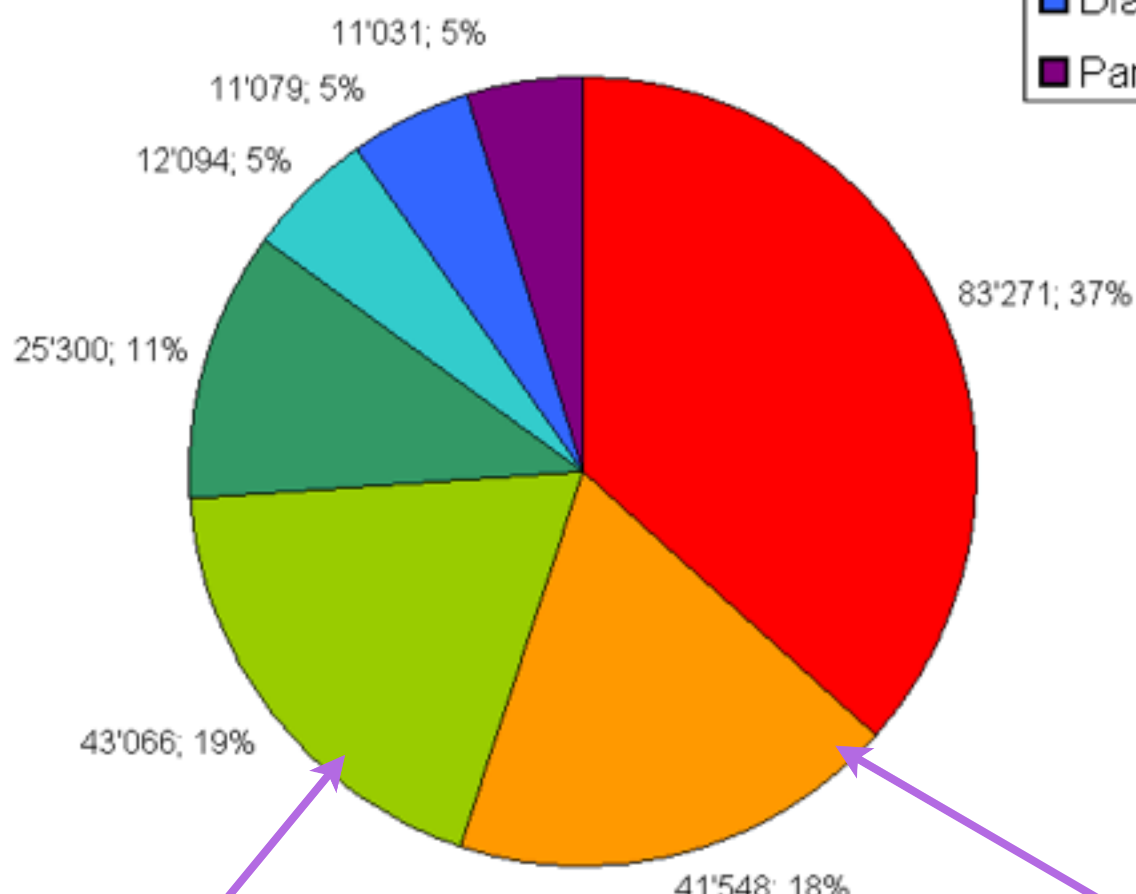
COSMO-OPCODE: a legacy code migration project

- monolithic Fortran 90 code
- 250,000 lines of code

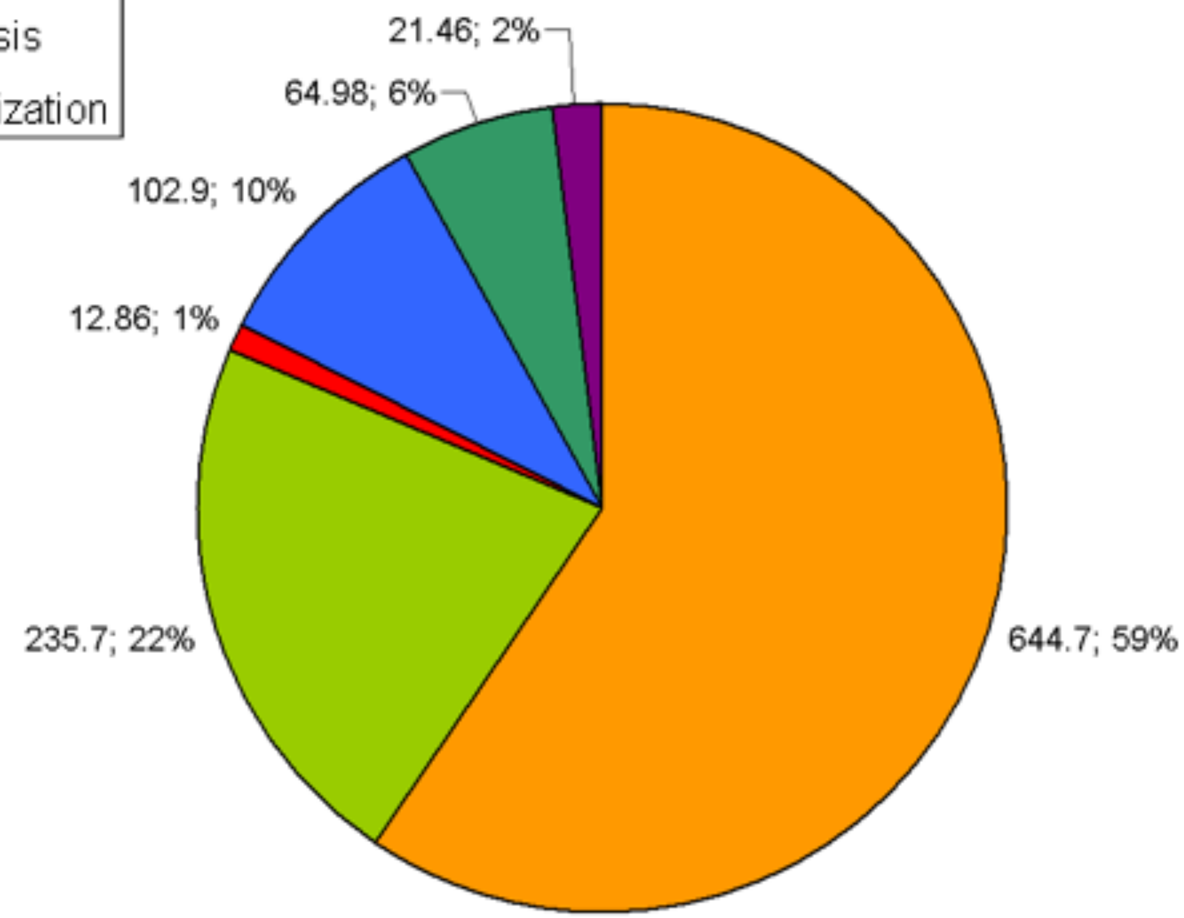
Runtime based 2 km production model of MeteoSwiss



% Code Lines (F90)



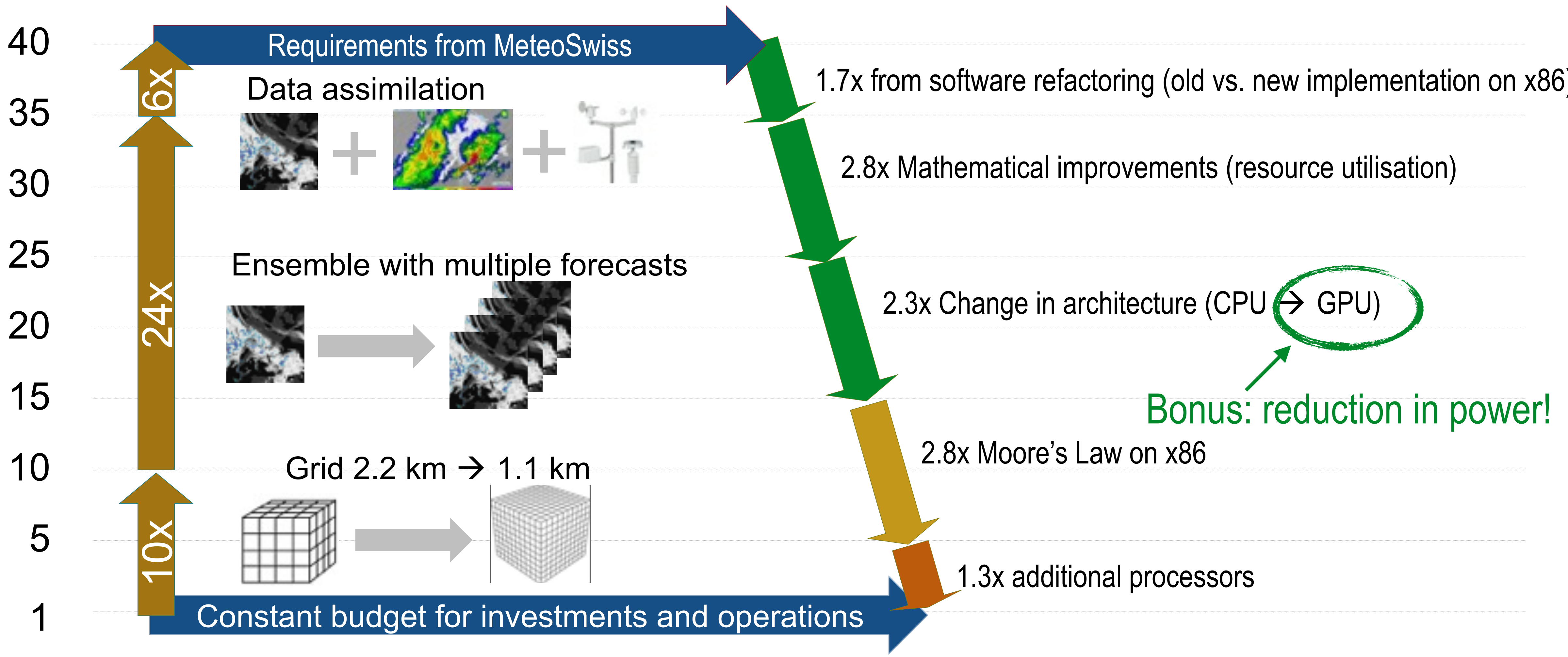
% Runtime



Original code (with OpenACC for GPU) Rewrite in C++ (with CUDA backend for GPU)

How we solved the problem

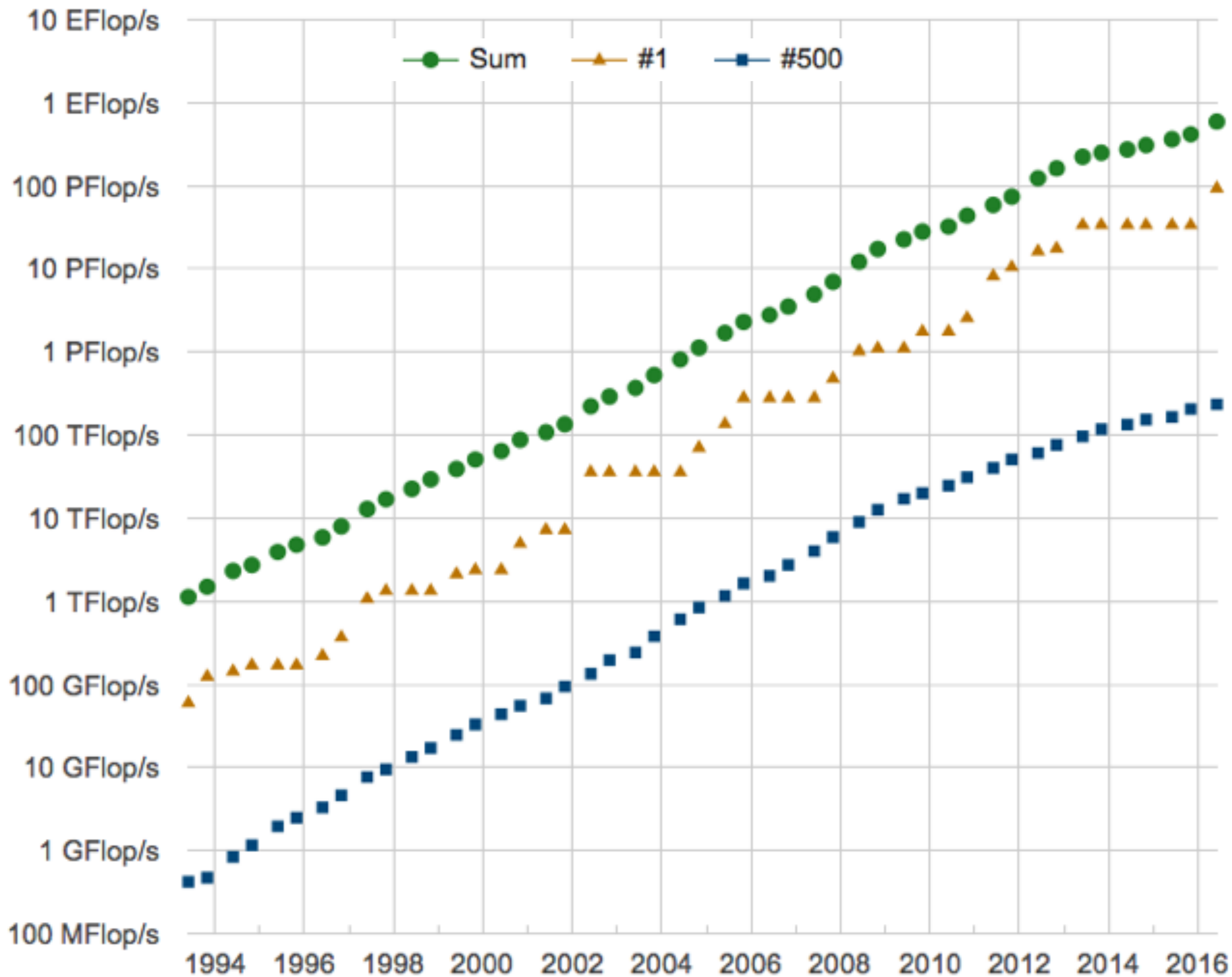
Investment in software allowed mathematical improvements and change in architecture



There is no silver bullet!

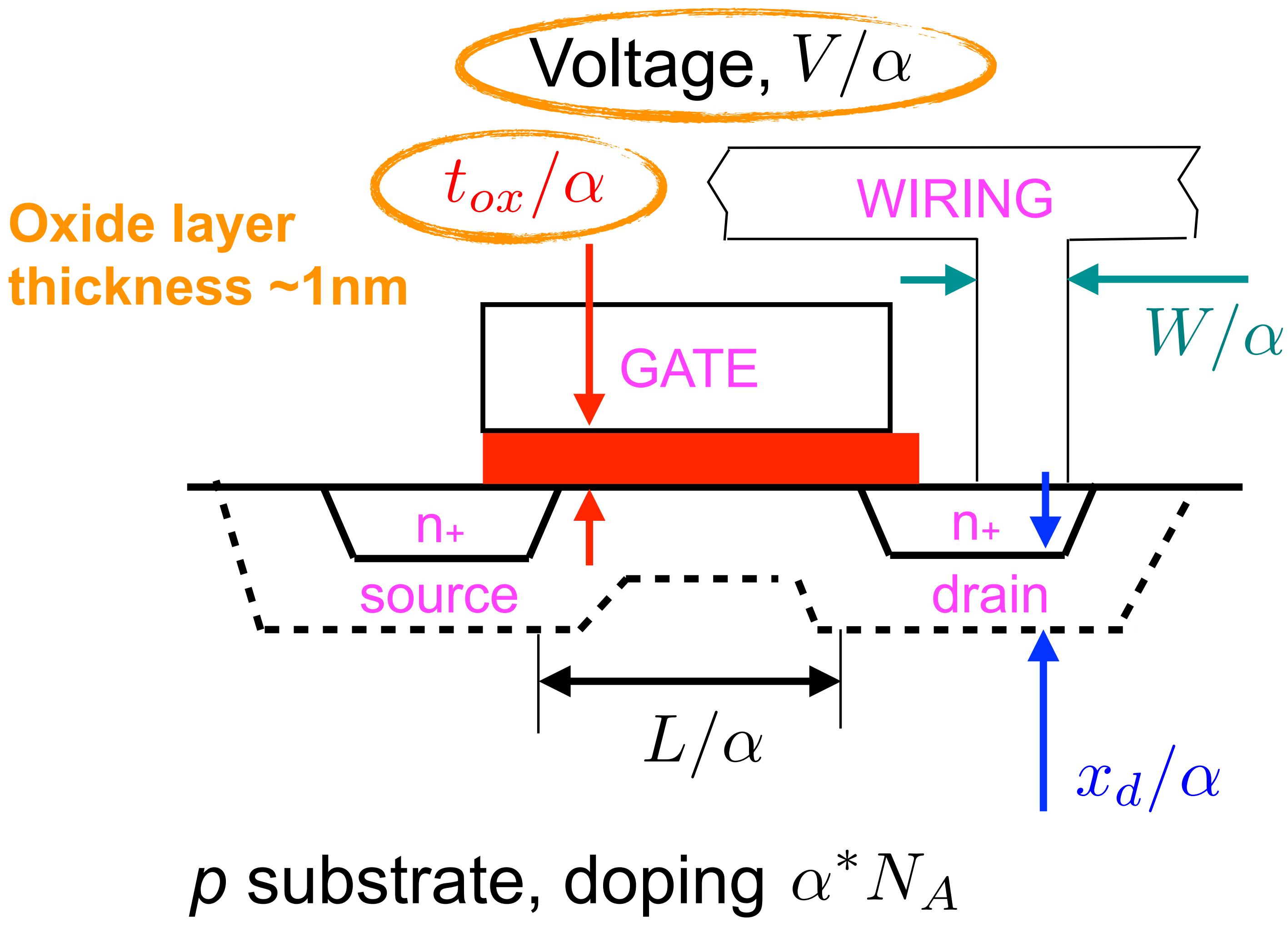
Why commodity processors?

Source: Andy Keane @ ISC’10


10 EFlop/s
1 EFlop/s
100 PFlop/s
10 PFlop/s
1 PFlop/s
100 TFlop/s
10 TFlop/s
1 TFlop/s
100 GFlop/s
10 GFlop/s
1 GFlop/s
100 MFlop/s
Sum
#1
#500
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016

The end of Denard Scaling

Robert H. Dennard (1974)



Source: Ronald Luijten, IBM-ZRL

SCALING

~~Voltage: V/α~~

~~Oxide: t_{ox}/α~~

Wire width: W/α

Gate Width: L/α

Diffusion: x_d/α

Substrate: $\alpha^* N_A$

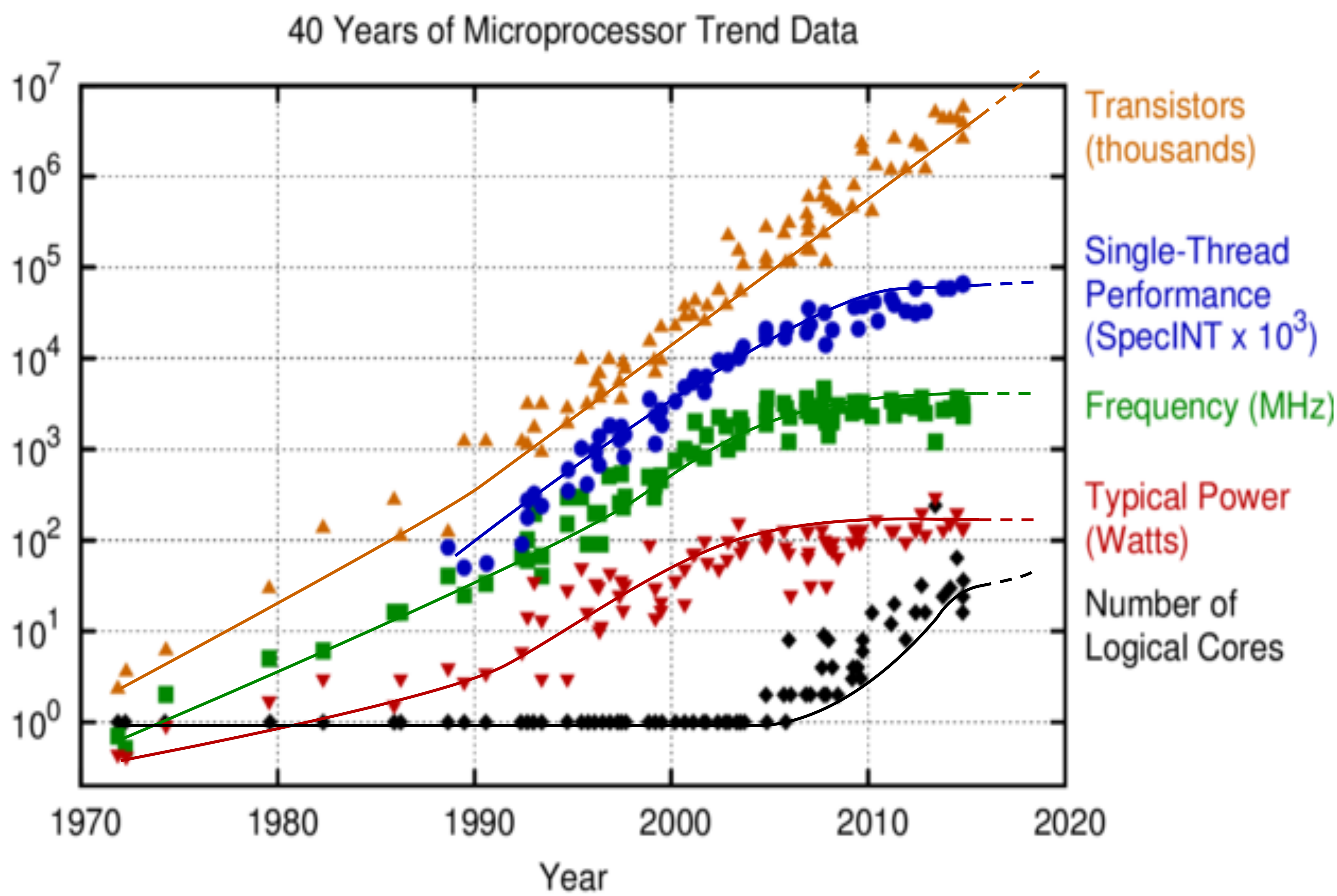
CONSEQUENCE:

Higher density: $\sim \alpha^2$

Higher speed: $\sim \alpha$

Power/ckt: $\sim 1/\alpha^2$

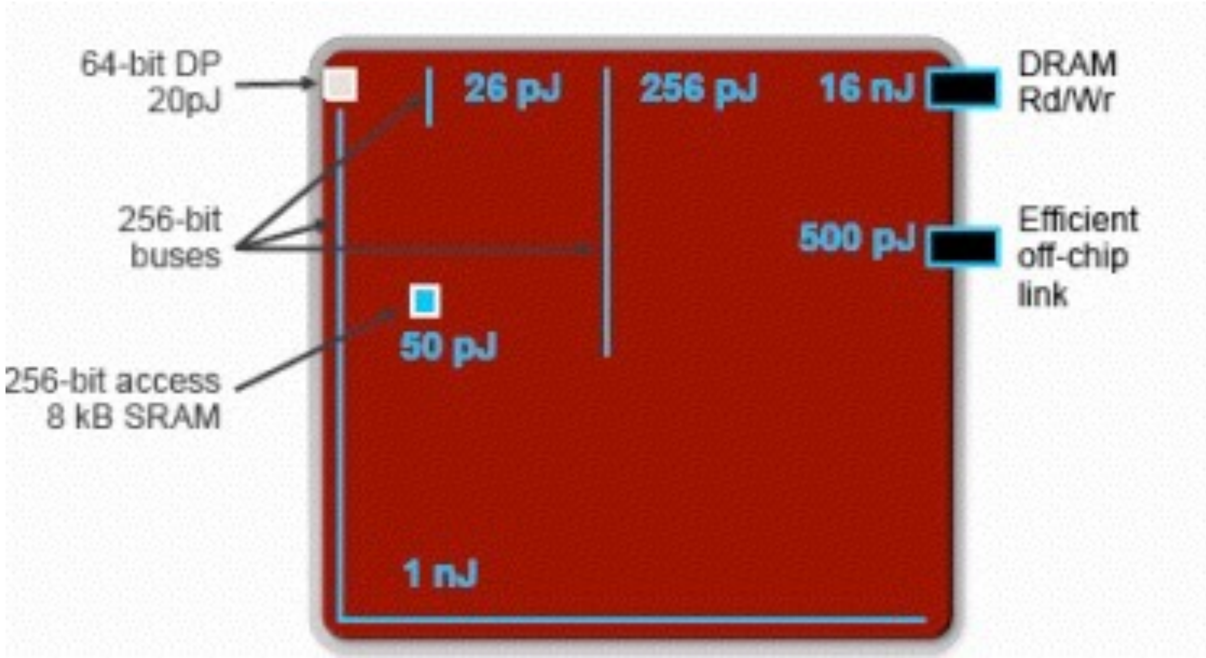
~~Power density: $\sim$ constant~~



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Who consumes how much energy on a 28nm processor

- 64 bit floating point unit: 20 pJ
- 256-bit access 8kB SRAM: 50 pJ
- 256-bit bus across die: 1,000 pJ
- Read/write to DRAM: 16,000 pJ

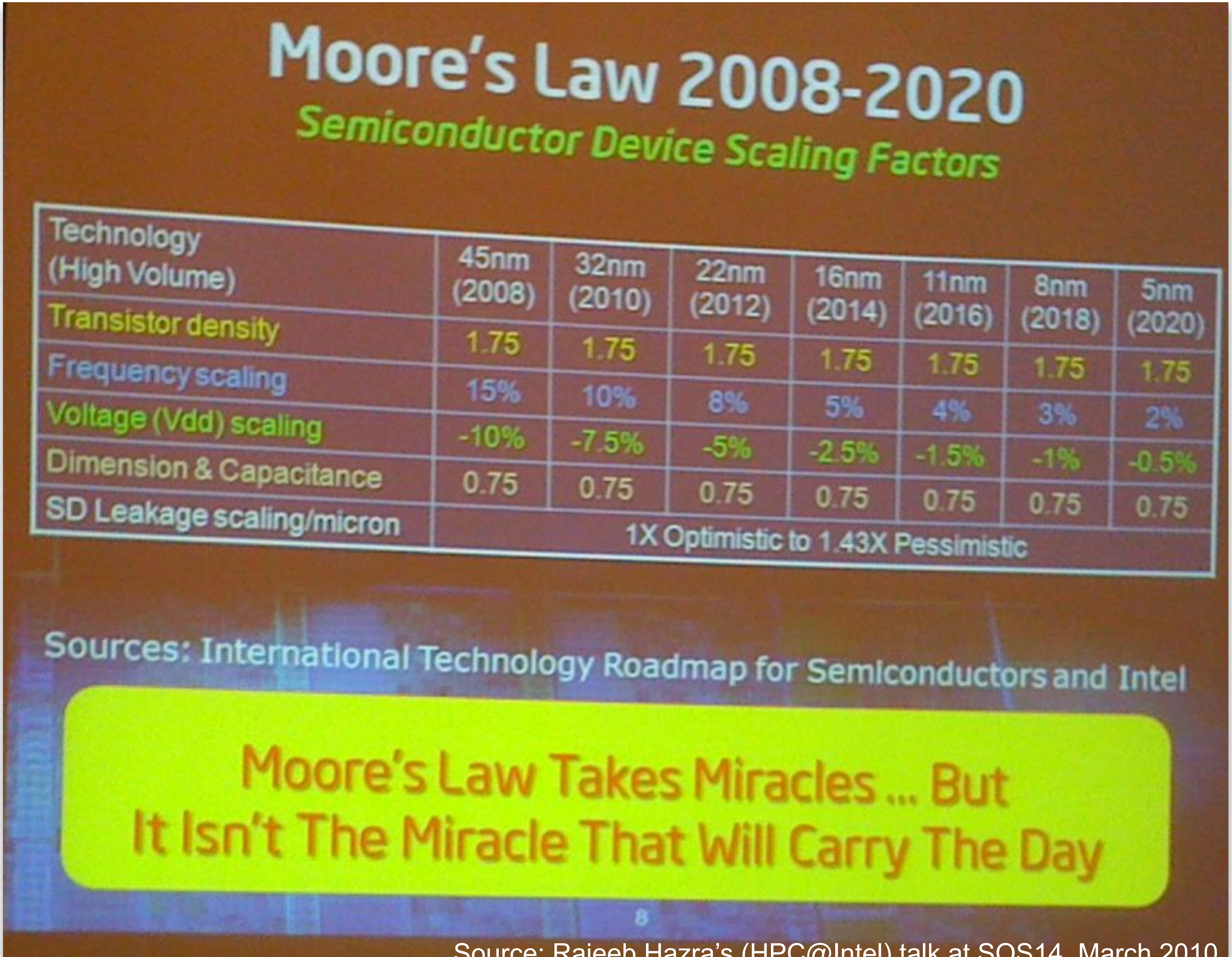


Source: Bill Dally, 2011

By a wide margin, most energy is spend in moving data on the die and to memory

Developing algorithms that maximise data locality should be THE TOP PRIORITY

20 mm



Technology (High Volume)	45nm (2008)	32nm (2010)	22nm (2012)	16nm (2014)	11nm (2016)	8nm (2018)	5nm (2020)
Transistor density	1.75	1.75	1.75	1.75	1.75	1.75	1.75
Frequency scaling	15%	10%	8%	5%	4%	3%	2%
Voltage (Vdd) scaling	-10%	-7.5%	-5%	-2.5%	-1.5%	-1%	-0.5%
Dimension & Capacitance	0.75	0.75	0.75	0.75	0.75	0.75	0.75
SD Leakage scaling/micron	1X Optimistic to 1.43X Pessimistic						

Source: Rajeeb Hazra's (HPC@Intel) talk at SOS14, March 2010

“Piz Daint”

Latency (µs)	Bandwidth (GB/s)	
Aries chip	1.28	9.89
Within Chassis	1.61	9.65
Between Chassis	1.58	9.82
Between Groups	2.53	9.63

Cray XC30 with 5272 hybrid, GPU accelerated compute nodes

Compute node:
> Host: Intel Xeon E5 2670 (SandyBridge 8c) ← 2.5 GHz (~0.38 ns)
> Accelerator: One NVIDIA K20X GPU (GK110) ← 0.73 GHz (~1.4ns)

Architectural diversity is here to stay, because it is a consequence of the dusk of CMOS scaling (Moore’s Law)

What are the implications?

Complexity in software is one, but we don’t understand all implications

Physics of the computer matters more than ever

velocities

$$\frac{\partial u}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\frac{\partial E_h}{\partial \lambda} - vV_a\right\} - \dot{\zeta}\frac{\partial u}{\partial \zeta} - \frac{1}{\rho a\cos\varphi}\left(\frac{\partial p'}{\partial \lambda} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \lambda}\frac{\partial p'}{\partial \zeta}\right) + M_u$$

$$\frac{\partial v}{\partial t} = -\left\{\frac{1}{a}\frac{\partial E_h}{\partial \varphi} + uV_a\right\} - \dot{\zeta}\frac{\partial v}{\partial \zeta} - \frac{1}{\rho a}\left(\frac{\partial p'}{\partial \varphi} - \frac{1}{\sqrt{\gamma}}\frac{\partial p_0}{\partial \varphi}\frac{\partial p'}{\partial \zeta}\right) + M_v$$

$$\frac{\partial w}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial w}{\partial \lambda} + v\cos\varphi\frac{\partial w}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial w}{\partial \zeta} + \frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial p'}{\partial \zeta} + M_w + g\frac{\rho_0}{\rho}\left\{\frac{(T-T_0)}{T} - \frac{T_0 p'}{T p_0} + \left(\frac{R_v}{R_d} - 1\right)q^v - q^l - q^f\right\}$$

pressure

$$\frac{\partial p'}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial p'}{\partial \lambda} + v\cos\varphi\frac{\partial p'}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial p'}{\partial \zeta} + g\rho_0 w - \frac{c_{pd}}{c_{vd}}pD$$

temperature

$$\frac{\partial T}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial T}{\partial \lambda} + v\cos\varphi\frac{\partial T}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial T}{\partial \zeta} - \frac{1}{\rho c_{vd}}pD + Q_T$$

water

$$\frac{\partial q^v}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^v}{\partial \lambda} + v\cos\varphi\frac{\partial q^v}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial q^v}{\partial \zeta} - (S^l + S^f) + M_{q^v}$$

$$\frac{\partial q^{l,f}}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial q^{l,f}}{\partial \lambda} + v\cos\varphi\frac{\partial q^{l,f}}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial q^{l,f}}{\partial \zeta} - \frac{g}{\sqrt{\gamma}}\frac{\rho_0}{\rho}\frac{\partial P_{l,f}}{\partial \zeta} + S^{l,f} + M_{q^{l,f}}$$

turbulence

$$\frac{\partial e_t}{\partial t} = -\left\{\frac{1}{a\cos\varphi}\left(u\frac{\partial e_t}{\partial \lambda} + v\cos\varphi\frac{\partial e_t}{\partial \varphi}\right)\right\} - \dot{\zeta}\frac{\partial e_t}{\partial \zeta} + K_m^v\frac{g\rho_0}{\sqrt{\gamma}}\left\{\left(\frac{\partial u}{\partial \zeta}\right)^2 + \left(\frac{\partial v}{\partial \zeta}\right)^2\right\} + \frac{g}{\rho\theta_v}F^{\theta_v} - \frac{\sqrt{2}e_t^{3/2}}{\alpha_M l} + M_{e_t}$$

Timestep

implicit (sparse) — explicit (RK3) — implicit (sparse solver) — explicit (leapfrog)

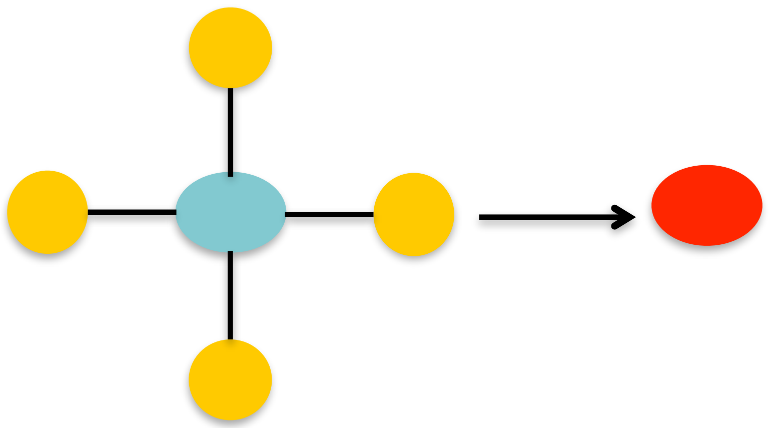
1x **physics et al. tendencies**

3x **horizontal adv.** **vertical adv.** ~10x **fast wave solver**

water adv.

Stencil example: Laplace operator in 2D

```
lap(i,j,k) = -4.0 * data(i,j,k) +
    data(i+1,j,k) + data(i-1,j,k) +
    data(i,j+1,k) + data(i,j-1,k);
```



Two main components of an operator on a structured grid

1. Loop-logic defines stencil application domain and order
2. Stencil defines the operator to be applied

```
do k = kstart, kend
  do j = jstart, jend
    do i = istart, iend
      lap(i, j, k) = -4.0 * data(i,   j  , k) + &
        data(i+1, j, , k) + data(i-1, j  , k) + &
        data(i  , j+1, k) + data(i  , j-1, k)
     end do
  end do
end do
```

```
enum { data, lap };

template<typename TEnv>
struct Laplace
{
  STENCIL_STAGE(Tenv)
  STAGE_PARAMETER(FullDomain, data)
  STAGE_PARAMETER(FullDomain, lap)

  static void Do()
  {
    lap::Center() =
      -4.0 * data::Center() +
      data::At(iplus1) +
      data::At(iminus1) +
      data::At(jplus1) +
      data::At(jminus1);
  }
};
```

```
IJKRealField lapfield, datafield;
Stencil stencil;

StencilCompiler::Build(
pack_parameters(
    Param<lap, cInOut>(lapfield),
    Param<data, cIn>(datafield)
),
  concatenate_sweeps(
    define_sweep<KLoopFullDomain>(
      define_stages(
        StencilStage<Laplace, IJRangeComplete>()
      )
    )
  )
);

stencil.Apply();
```

Stencil

```
enum { data, lap };

template<typename TEnv>
struct Laplace
{
  STENCIL_STAGE(Tenv)
  STAGE_PARAMETER(FullDomain, data)
  STAGE_PARAMETER(FullDomain, lap)

  static void Do()
  {
    lap::Center() =
      -4.0 * data::Center() +
      data::At(iplus1) +
      data::At(iminus1) +
      data::At(jplus1) +
      data::At(jminus1);
  }
};
```

Loop logic

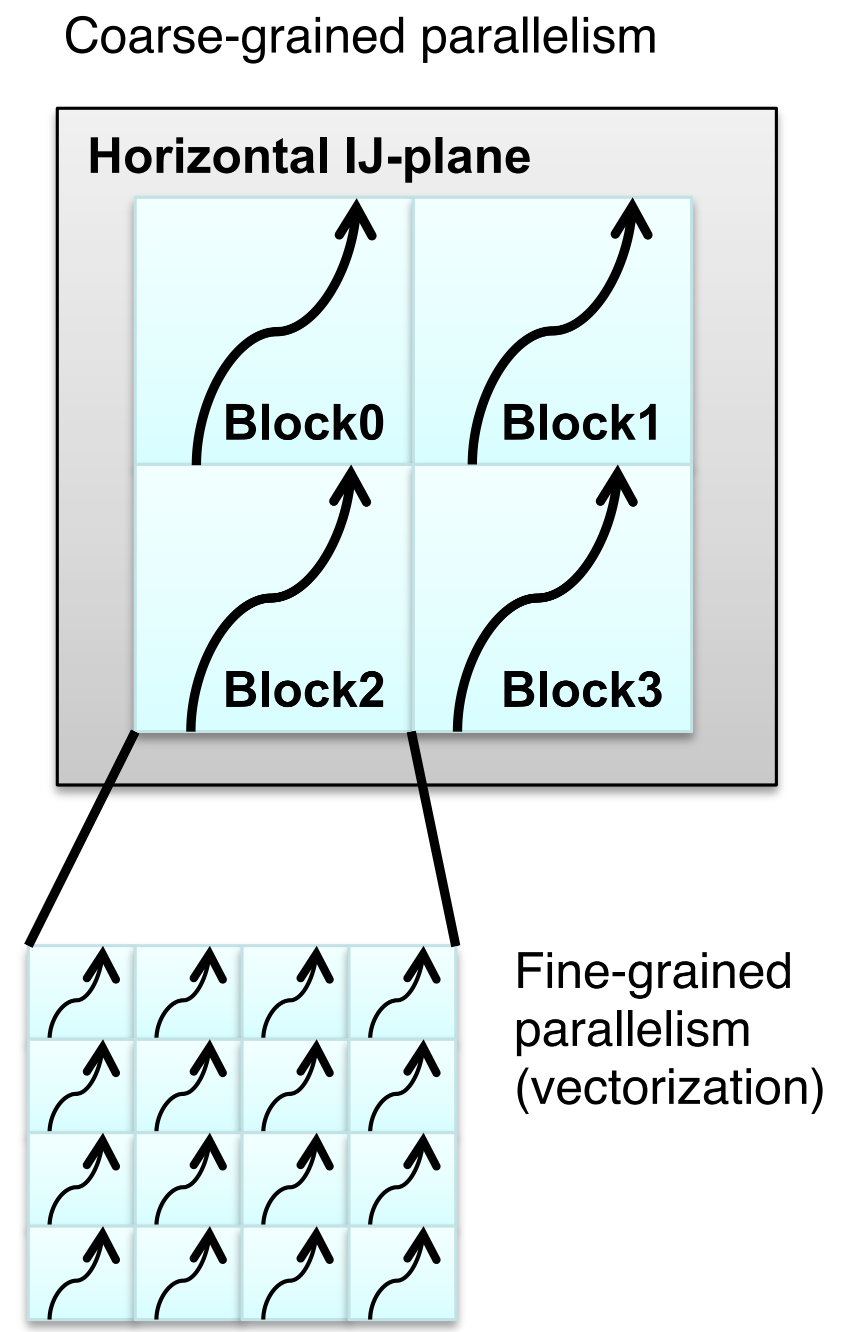
```
IJKRealField lapfield, datafield;
Stencil stencil;

StencilCompiler::Build(
pack_parameters(
    Param<lap, cInOut>(lapfield),
    Param<data, cIn>(datafield)
),
  concatenate_sweeps(
    define_sweep<KLoopFullDomain>(
      define_stages(
        StencilStage<Laplace, IJRangeComplete>()
      )
    )
  )
);

stencil.Apply();
```

Architecture dependent backend

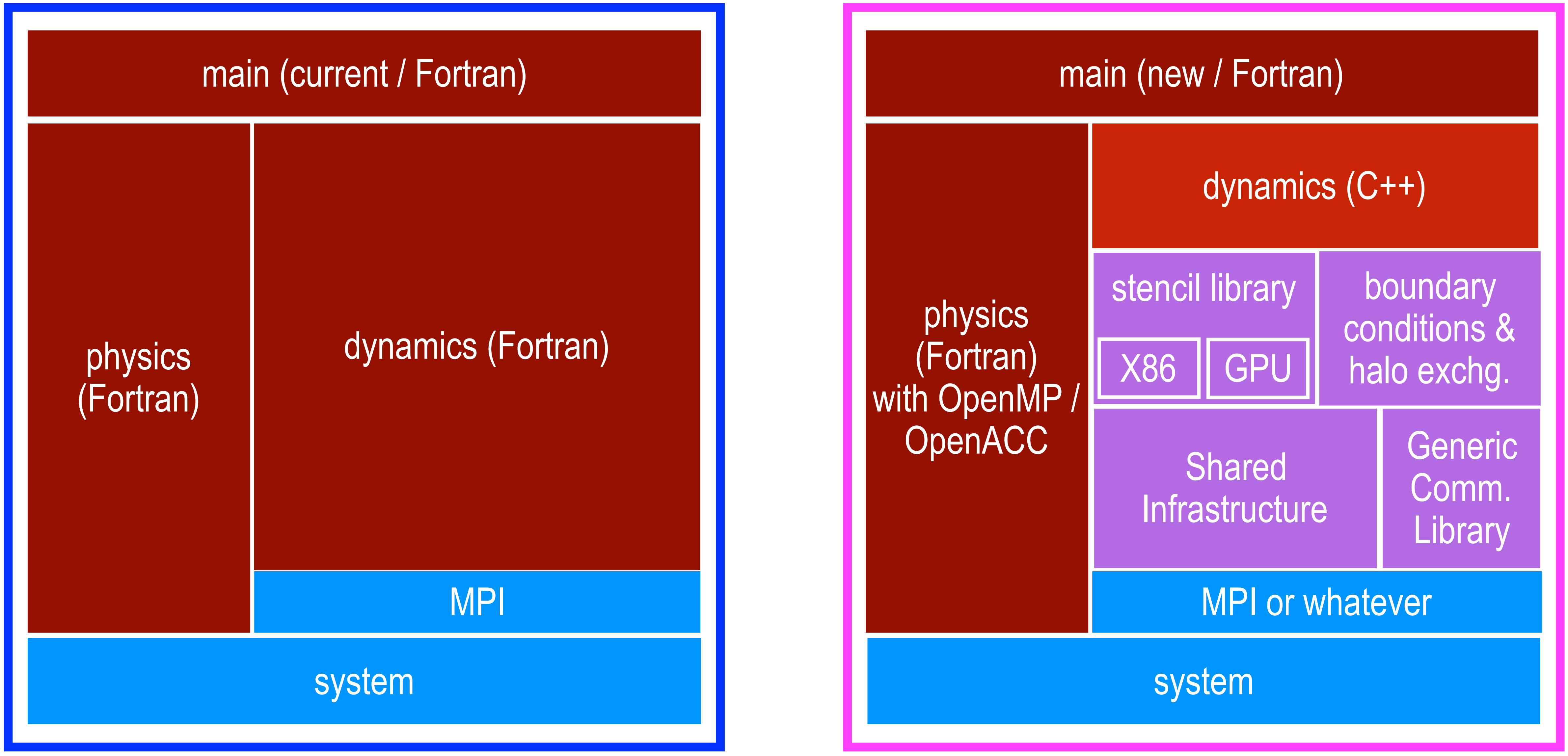
- The same user-level code can be compiled with different, architecture dependent backends
- **multi-core CPU (x86) – SIMD**
  - kij-storage
  - ij-blocking
  - Coarse: OpenMP theads
  - Fine: vectorisation by compiler
- **GPU (Tesla) – SIMT**
  - ijk-storage
  - Coarse: CUDA thread blocks
  - Fine: CUDA threads
  - software managed caching



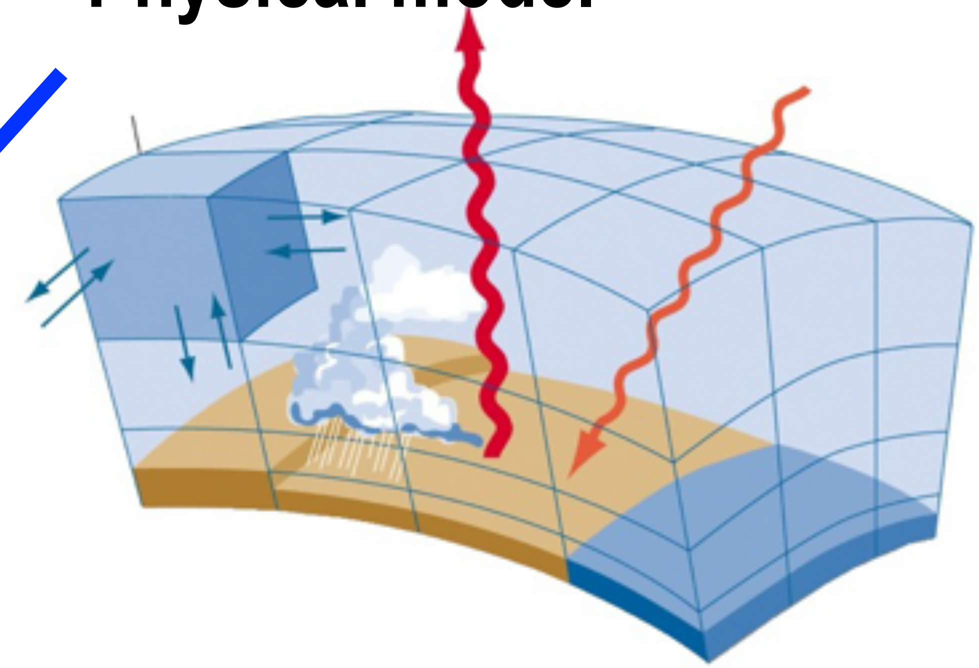
References and Collaborators

- Peter Messmer and his team at the NVIDIA co-design lab at ETH Zurich
- Teams at CSCS and Meteo Suisse, group of Christoph Schaer @ ETH Zurich
- O. Fuhrer, C. Osuna, X. Lapillonne, T. Gysi, B. Cumming, M. Bianco, A. Arteaga, T. C. Schulthess, "**Towards a performance portable, architecture agnostic implementation strategy for weather and climate models**", Supercomputing Frontiers and Innovations, vol. 1, no. 1 (2014), see superfri.org
- G. Fourestey, B. Cumming, L. Gilly, and T. C. Schulthess, "**First experience with validating and using the Cray power management database tool**", Proceedings of the Cray Users Group 2014 (CUG14) (see arxiv.org for reprint)
- B. Cumming, G. Fourestey, T. Gysi, O. Fuhrer, M. Fatica, and T. C. Schulthess, "**Application centric energy-efficiency study of distributed multi-core and hybrid CPU-GPU systems**", Proceedings of the International Conference on High-Performance Computing, Networking, Storage and Analysis, SC'14, New York, NY, USA (2014). ACM
- T. Gysi, C. Osuna, O. Fuhrer, M. Bianco and T. C. Schulthess, "**STELLA: A domain-specific tool for structure grid methods in weather and climate models**", to be published in Proceedings of the International Conference on High-Performance Computing, Networking, Storage and Analysis, SC'15, New York, NY, USA (2015). ACM

COSMO: old and new (refactored) code



Physical model



Wind $\rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$

Pressure $\dot{p} = -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$

Temperature $\rho c_{pd}\dot{T} = \dot{p} + Q_h$

Water $\rho\dot{q}^{v} = -\nabla\cdot\mathbf{F}^{v} - (I^{l} + I^{f})$

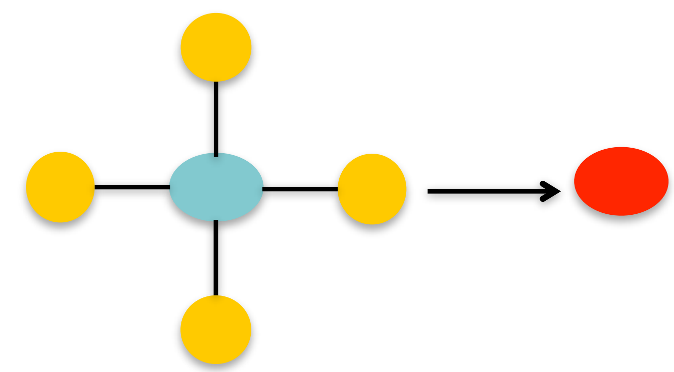
$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

Density $\rho = p[R_d\,(1+(R_v/R_d-1)\,q^{v}-q^{l}-q^{f})\,T]^{-1}$

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
lap(i,j,k) = -4.0 * data(i,j,k) +
    data(i+1,j,k) + data(i-1,j,k) +
    data(i,j+1,k) + data(i,j-1,k);
```

Imperative code

Compilation

Computer

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

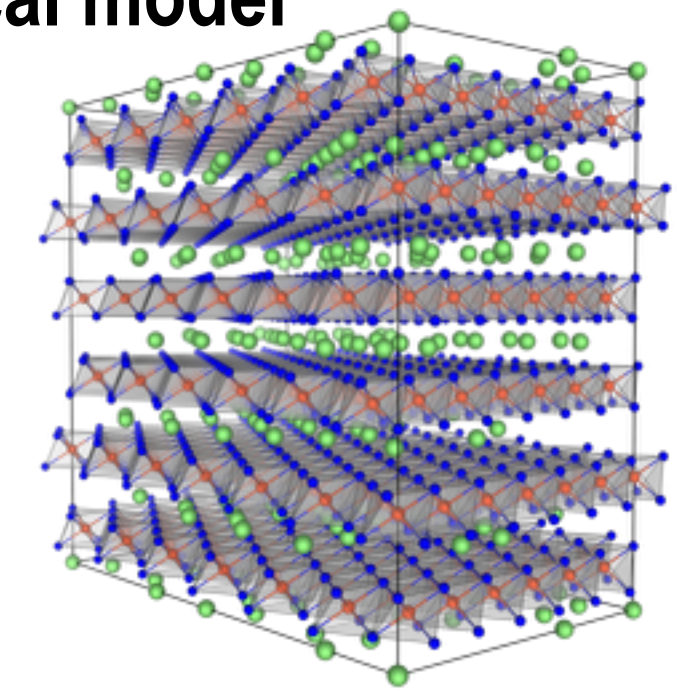
DFT ground state

Eigen-value problem $\hat{H}\Psi_{i\mathbf{k}}(\mathbf{r}) = \epsilon_{i\mathbf{k}}\Psi_{i\mathbf{k}}(\mathbf{r})$

Charge density $\rho_{\sigma\sigma'}(\mathbf{r}) = \sum_{i\mathbf{k}}^{occ} \Psi_{i\mathbf{k}}^{*\sigma'}(\mathbf{r})\Psi_{i\mathbf{k}}^{\sigma}(\mathbf{r})$

Effective potential and magnetic field $v^{eff}(\mathbf{r}) = v^{eff}[\rho_{\sigma\sigma'}(\mathbf{r})](\mathbf{r})$ $\mathbf{B}^{eff}(\mathbf{r}) = \mathbf{B}^{eff}[\rho_{\sigma\sigma'}(\mathbf{r})](\mathbf{r})$

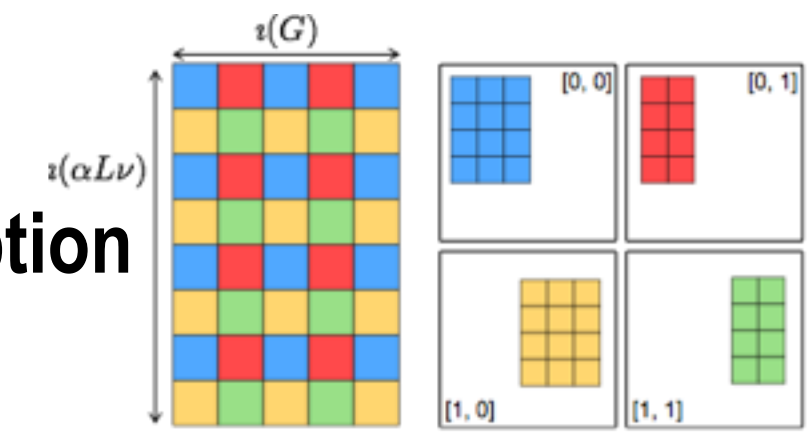
Physical model



Mathematical description

Domain science & applied mathematics

Algorithmic description



Imperative code

Compilation

Computer

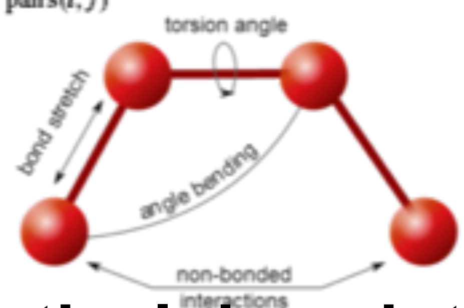
Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

$$V(r) = \sum_{\text{bonds}} k_b(b-b_0)^2 + \sum_{\text{angles}} k_\theta(\theta-\theta_0)^2 + \sum_{\text{dihedrals}} k_\phi(1+\cos(n\phi-\phi_0)) + \sum_{\text{impropers}} k_\psi(\psi-\psi_0)^2 + \sum_{\substack{\text{non-bonded}\\ \text{pairs}(i,j)}} 4\varepsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] + \sum_{\substack{\text{non-bonded}\\ \text{pairs}(i,j)}} \frac{q_i q_j}{\varepsilon_D r_{ij}}$$

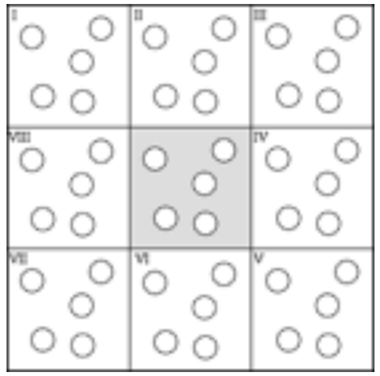


Physical model

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
Cray PAT API "11 loop init mtran" in UKCA_CDAGWITHNUCL
      call pat_region_begin(11,'11 loop init mtran',pat_stat)
!
!!!       DO IMODE=1,NMODES
!!!        NDOLD(:,IMODE)=ND(:,IMODE)
!!!        DO ICP=1,NCP
!!!         MDOLD(:,IMODE,ICP)=MD(:,IMODE,ICP)
!!!         DO JMODE=1,NMODES
!!!          MTRAN(:,IMODE,JMODE,ICP)=0.0
!!!         ENDDO
!!!        ENDDO
!!!       ENDDO
!
! replace triple loops above with F90 array syntax (let
compiler decide)
      NDOLD=ND
      MDOLD=MD
      MTRAN=0.0
      call pat_region_end(11,pat_stat)
```

Imperative code

Compilation

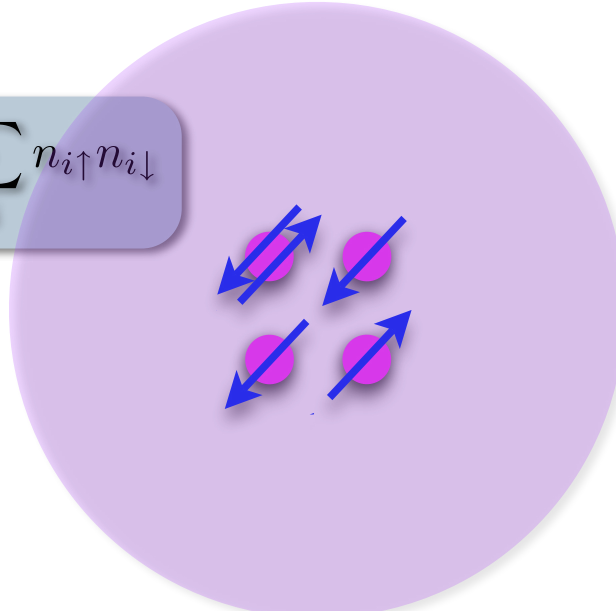
Computer

Computer engineering

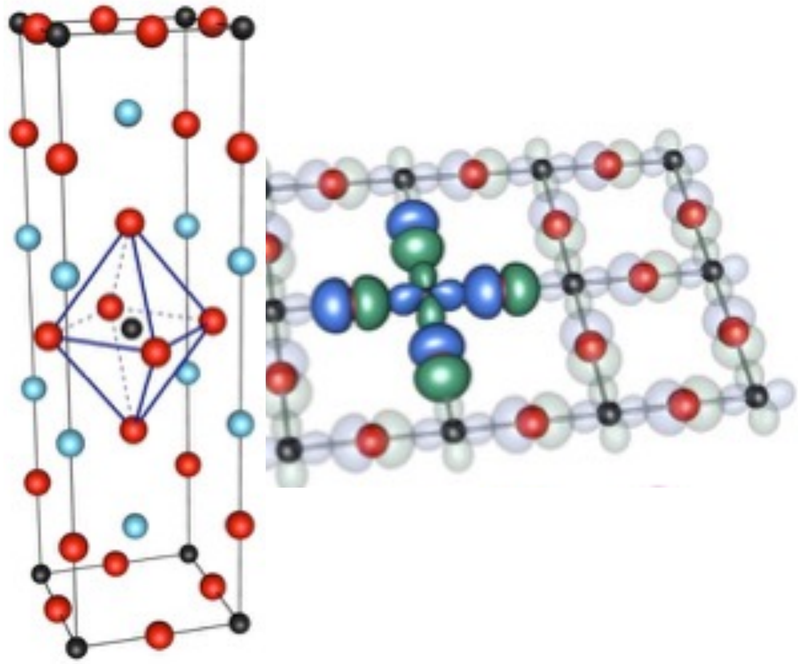
Schulthess, Nature Physics, vol 11, 369-373 (2015)

ETH zürich

$$\mathcal{H} = -t \sum_{\langle ij \rangle, \sigma} c^\dagger_{i\sigma} c_{j\sigma} + U \sum_i n_{i\uparrow} n_{i\downarrow}$$



Physical model

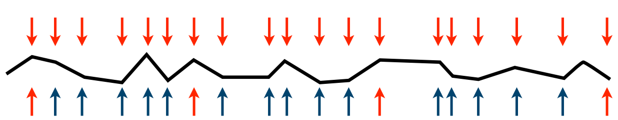


Mathematical description

Domain science & applied mathematics

Algorithmic description

$$\mathbf{G}_c(\{s_i, l\}_{k+1}) = \mathbf{G}_c(\{s_i, l\}_k) + \mathbf{a}_k \times \mathbf{b}^t_k$$



$$\mathbf{G}_c(\{s_i, l\}_{k+1}) = \mathbf{G}_c(\{s_i, l\}_0) + [\mathbf{a}_0|\mathbf{a}_1|...|\mathbf{a}_k] \times [\mathbf{b}_0|\mathbf{b}_1|...|\mathbf{b}_k]^t$$

Imperative code

Compilation

Computer

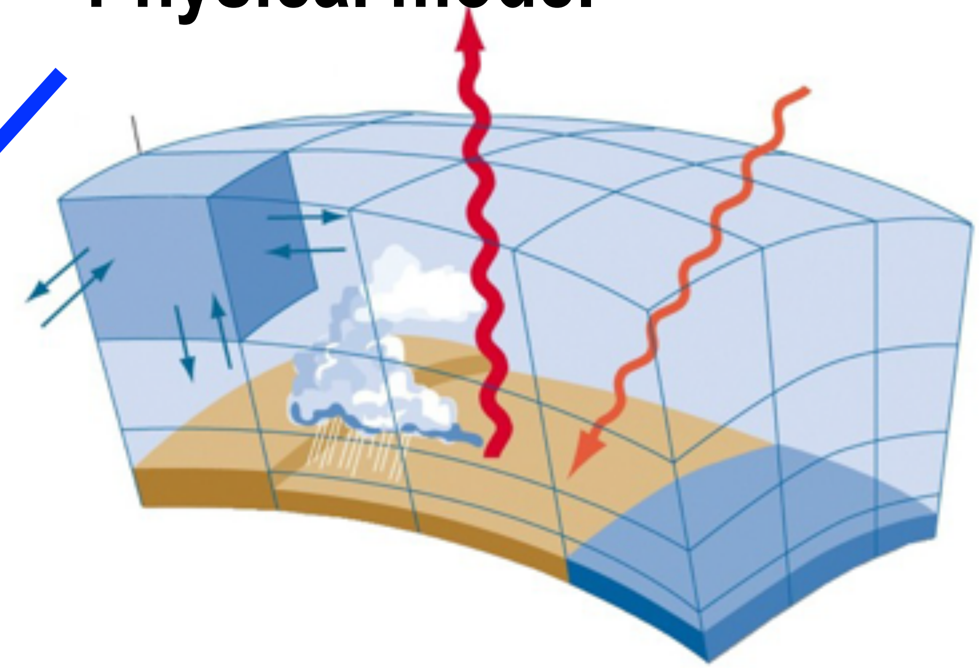
Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Physical model



Wind $\rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$

Pressure $\dot{p} = -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$

Temperature $\rho c_{pd}\dot{T} = \dot{p} + Q_h$

Water $\rho\dot{q}^v = -\nabla\cdot\mathbf{F}^v - (I^l + I^f)$

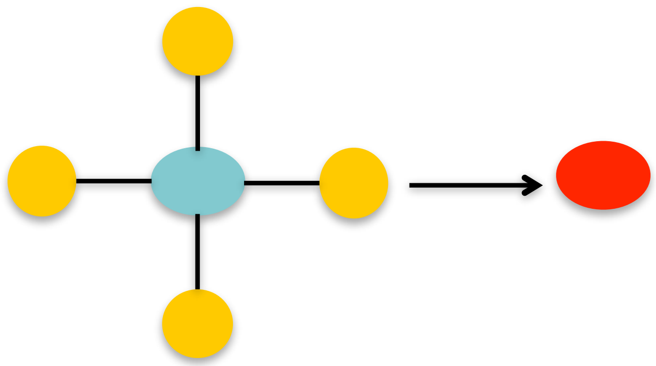
$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

Density $\rho = p[R_d\,(1+(R_v/R_d-1)\,q^v - q^l - q^f)\,T]^{-1}$

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
lap(i,j,k) = -4.0 * data(i,j,k) +
    data(i+1,j,k) + data(i-1,j,k) +
    data(i,j+1,k) + data(i,j-1,k);
```

Imperative code

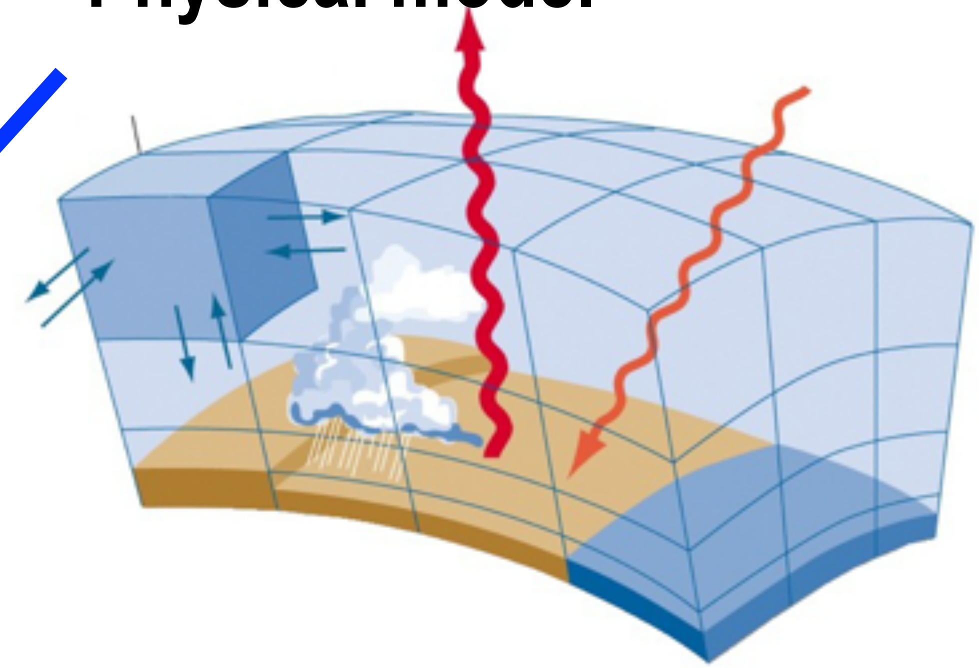
Compilation

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

ETH zürich

Physical model



$$\text{Wind } \rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$$

$$\text{Pressure } \dot{p} = -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$$

$$\text{Temperature } \rho c_{pd}\dot{T} = \dot{p} + Q_h$$

$$\text{Water } \rho\dot{q}^v = -\nabla\cdot\mathbf{F}^v - (I^l + I^f)$$

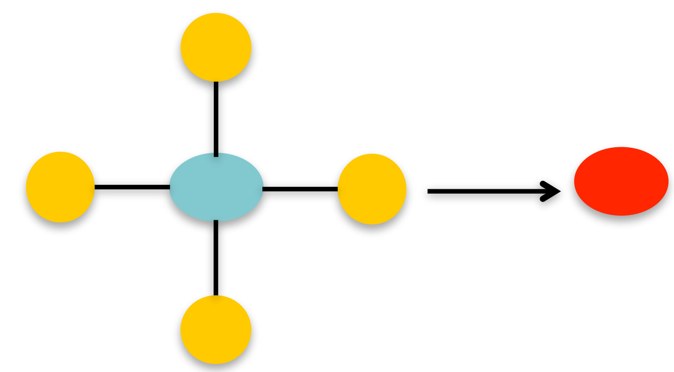
$$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$$

$$\text{Density } \rho = p[R_d(1+(R_v/R_d-1)q^v - q^l - q^f)T]^{-1}$$

Mathematical description

Domain science & applied mathematics

Algorithmic description



```
lap(i,j,k) = -4.0 * data(i,j,k) +
    data(i+1,j,k) + data(i-1,j,k) +
    data(i,j+1,k) + data(i,j-1,k);
```

Imperative code

Compilation

Computer engineering

Schulthess, Nature Physics, vol 11, 369-373 (2015)

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Wind $\rho\dot{\mathbf{v}} = -\nabla p + \rho\mathbf{g} - 2\Omega\times(\rho\mathbf{v}) + \mathbf{F}$

Pressure $\dot{p} = -(c_{pd}/c_{vd})\, p\nabla\cdot\mathbf{v} + (c_{pd}/c_{vd}-1)Q_h$

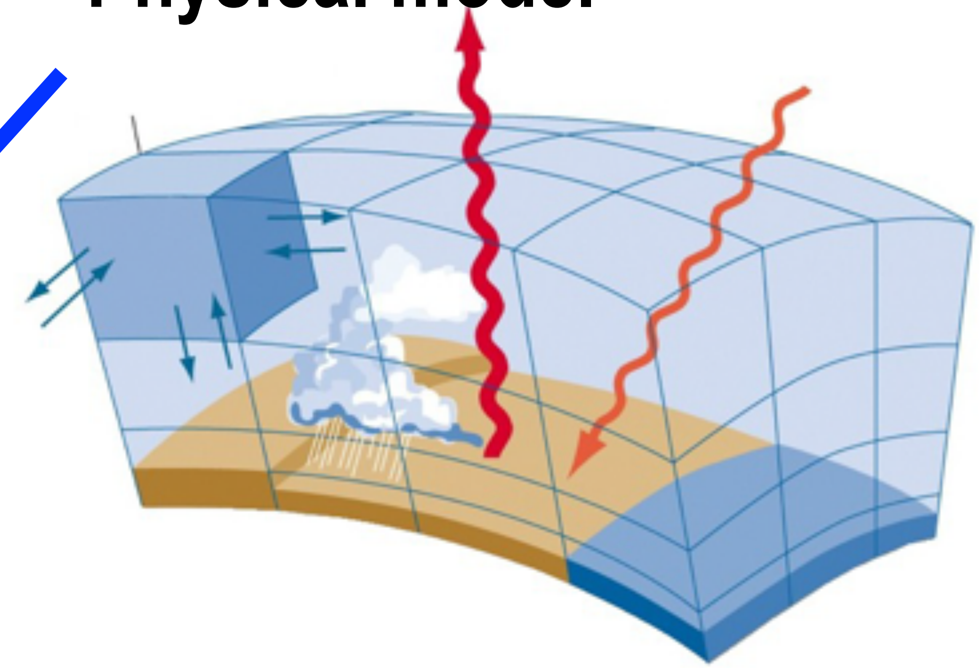
Temperature $\rho c_{pd}\dot{T} = \dot{p} + Q_h$

Water $\rho\dot{q}^{v} = -\nabla\cdot\mathbf{F}^{v} - (I^{l} + I^{f})$

$\rho\dot{q}^{l,f} = \nabla\cdot(\mathbf{P}^{l,f} + \mathbf{F}^{l,f}) + I^{l,f}$

Density $\rho = p[R_d\,(1+(R_v/R_d-1)\,q^{v} - q^{l} - q^{f})\,T]^{-1}$

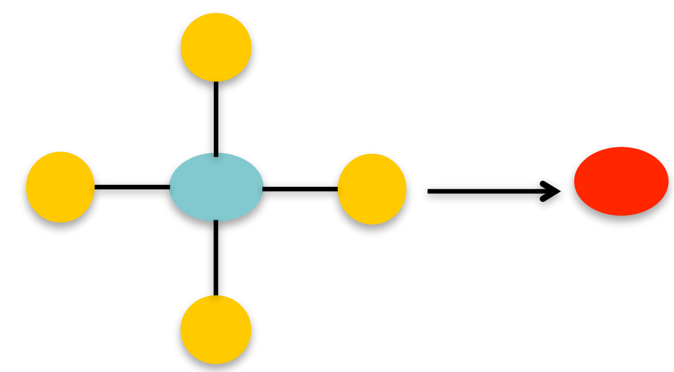
Physical model



Mathematical description

Domain science & applied mathematics

Algorithmic description



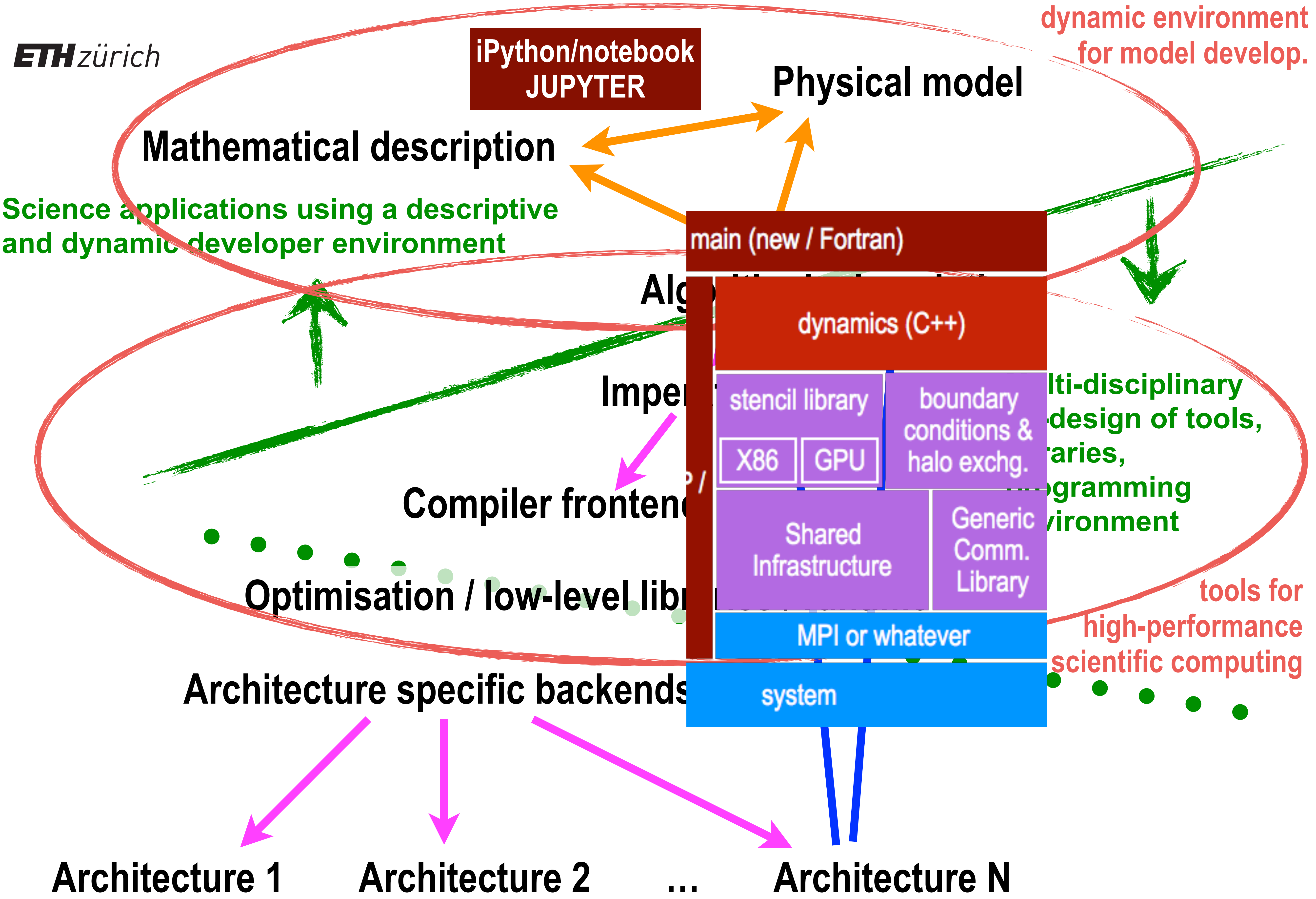
```
lap(i,j,k) = -4.0 * data(i,j,k) +
    data(i+1,j,k) + data(i-1,j,k) +
    data(i,j+1,k) + data(i,j-1,k);
```

Imperative code

Compilation

Computer engineering

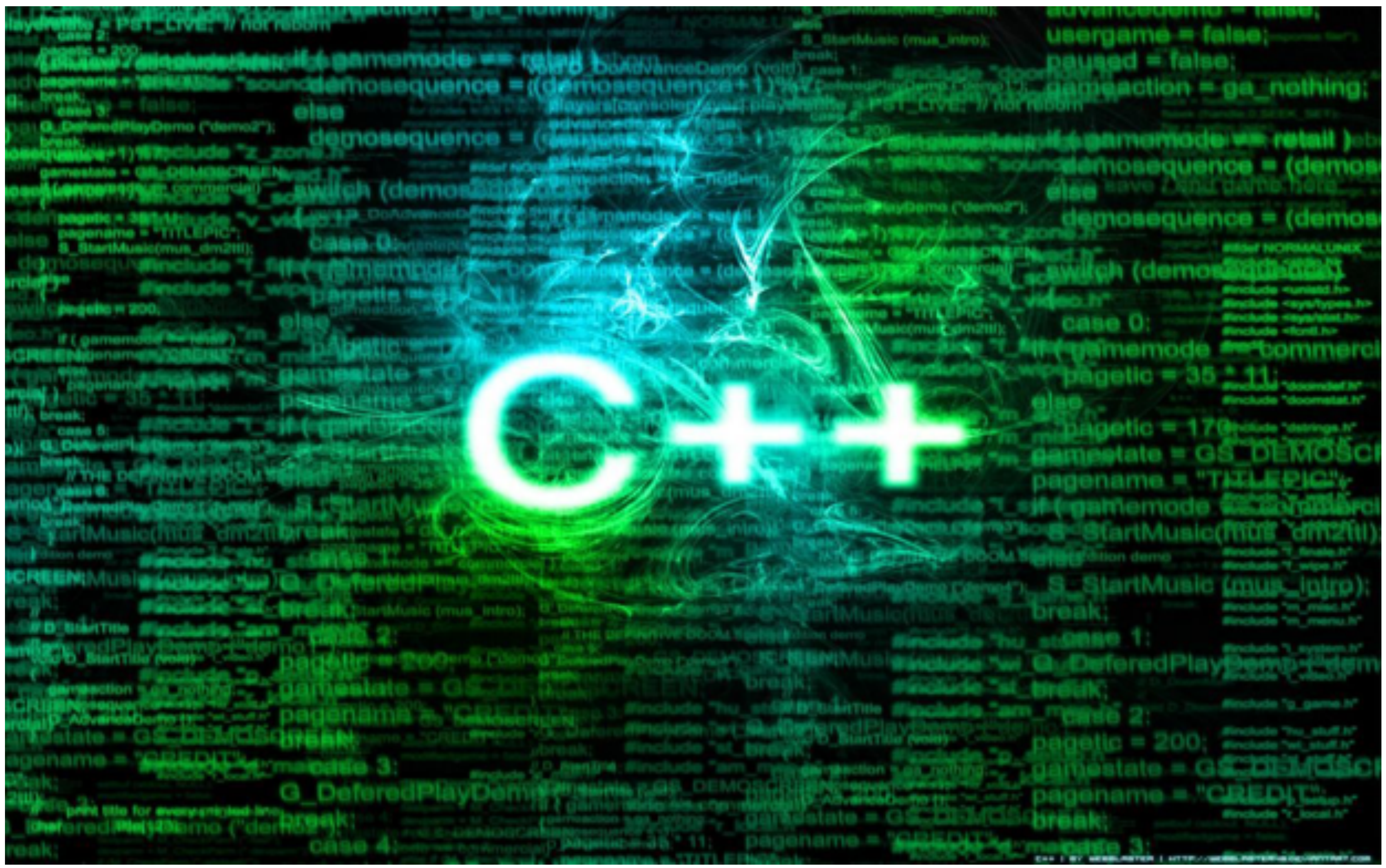
Schulthess, Nature Physics, vol 11, 369-373 (2015)



Schulthess, Nature Physics, vol 11, 369-373 (2015)

The good news

C++ standard is evolving quickly and implementations follow!



C++ 11, 14, HPX-3, … 17, 20, …

Who will pay for the implementation of Fortran, OpenACC, OpenMP, …?

for 8 GPUs, or $16k a piece

$500,000,000

$2,000,000,000

$13,000

Source: Andy Keane @ ISC'10

ETH zürich

sighpc

acm Association for Computing Machinery

PASC17

Platform for Advanced Scientific Computing Conference

Lugano Switzerland

26-28 June 2017


Call for Papers

The Platform for Advanced Scientific Computing (PASC) invites submissions for the PASC17 Conference, co-sponsored by the Association for Computing Machinery (ACM) and SIGHPC, which will be held at the Palazzo dei Congressi in Lugano, Switzerland, from June 26 to 28, 2017.

PASC17

Platform for Advanced Scientific Computing Conference

Lugano
Switzerland

26–28 June 2017

PASC17 is an interdisciplinary event in high performance computing that brings together domain science, applied mathematics and computer science - where computer science is focused on enabling the realization of scientific computation.

We are soliciting high-quality contributions of original research relating to high performance computing in eight domain-specific tracks:

- CLIMATE & WEATHER
- SOLID EARTH DYNAMICS
- LIFE SCIENCES
- CHEMISTRY & MATERIALS
- PHYSICS
- COMPUTER SCIENCE & APPLIED MATHEMATICS
- ENGINEERING
- EMERGING DOMAINS (SPECIAL TOPIC: PRECISION MEDICINE)

pasc17.pasc-conference.org

Areas of interest include (but are not limited to):

- The use of advanced computing systems for large-scale scientific applications
- Implementation strategies for science applications in energy-efficient computing architectures
- Domain-specific, languages, libraries or frameworks
- The integration of large-scale experimental and observational scientific data and high-performance data analytics and computing
- Best practices for sustainable software development and scientific application development

Committee Chairs

Jack Wells (Oak Ridge National Laboratory, USA)
Torsten Hoefler (ETH Zurich, Switzerland)

Submission Guidelines

We invite papers of 5-10 pages in length, which will be reviewed double blind. Full submissions guidelines can be found at www.pasc17.org.

- Submissions close: **12 December 2016**
- First review notification: **31 January 2017**
- Revised submissions close: **1 March 2017**
- Final acceptance notification: **11 April 2017**

Conference Participation and Proceedings

Accepted manuscripts will be **published in the ACM Digital Library** on the first day of the conference. Authors will be given 30-minute presentation slots at the conference, grouped in topically focused, parallel sessions.

Post-Conference Journal Submission

Following the conference, authors will have the **opportunity to develop their papers for publication in a relevant, computationally focused, domain-specific journal.**

Authors thus stand to benefit from the rapid and broad dissemination of results afforded by the conference venue and associated proceedings, and, from the impact associated with publication in a high-quality scientific journal.

Association for Computing Machinery

sighpc

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre