

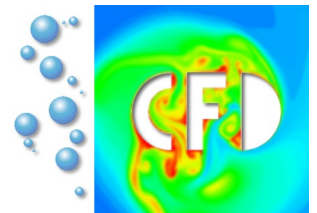
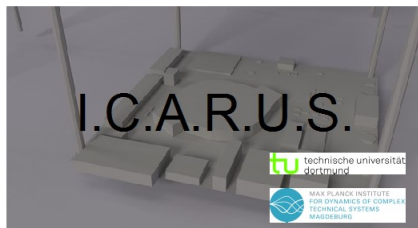
Performance engineering for hardware-, numerical- and energy-efficiency in FEM frameworks: modeling and control

Hardware-oriented Numerics in the FEAT software family

Markus Geveler

Higher-order DG methods and finite element software for modern architectures
Bath, 2016 / 6 / 1

markus.geveler@math.tu-dortmund.de
Institute f. Applied Mathematics, TU Dortmund



Outline

How can we take control of performance (different metrics) in FEM?

→ **Hardware-oriented Numerics**

- Our perspective on performance engineering in FEAT
- **Green computing, Hardware-oriented Numerics and Unconventional HPC**
- Simulation w.r.t. hardware-, numerical-, and energy-efficiency

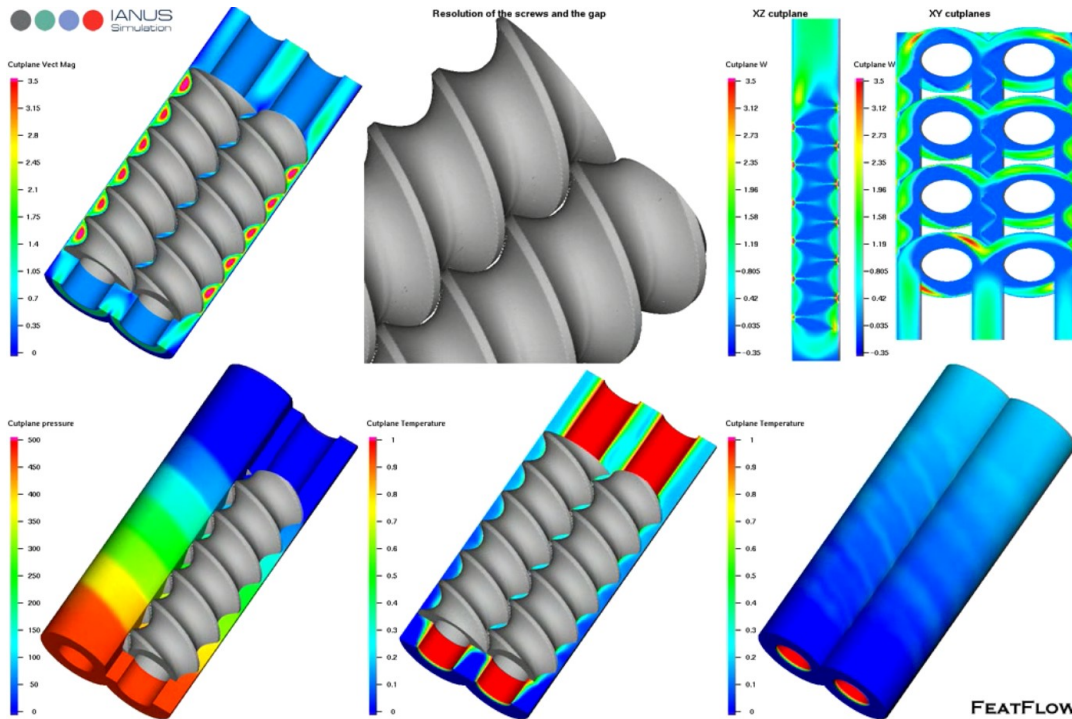
→ A case study for performance engineering in the **FEAT software family and SPPEXA**

→ **A prototype for future Data Centers**

- Preliminary work with ARM-based clusters
- **the I.C.A.R.U.S experimental cluster based on NVIDIA Tegra K1** combined with photovoltaic solar farming
- modeling energy in FEM frameworks

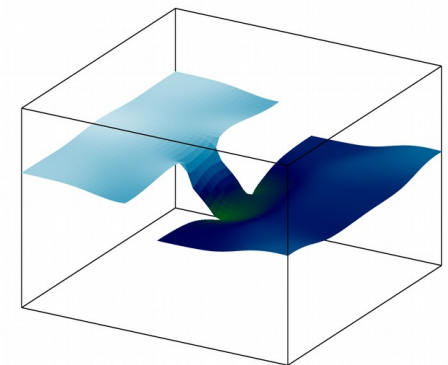
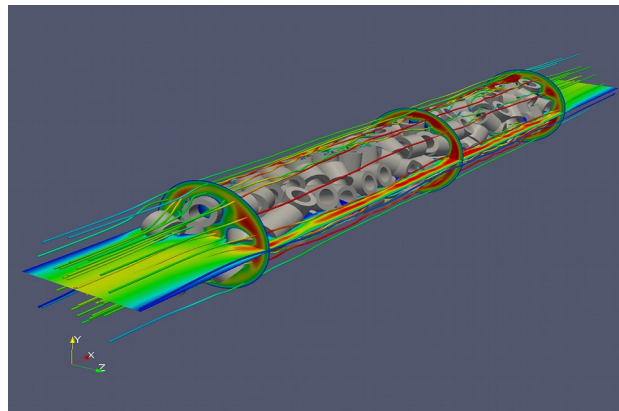
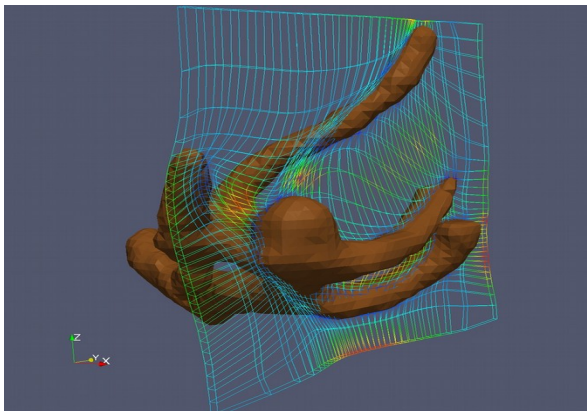
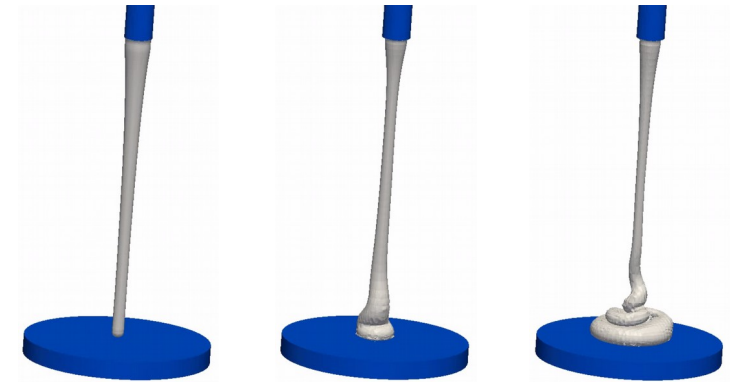
Motivation: where it all leads...

Simulation of technical flows with FEATFLOW



characteristics:

- high end modeling, numerics
- huge requirements: computation
- at some point: huge system to solve



Hardware

today's HPC facilities

- comprise heterogeneous compute nodes
- multicore CPU(s) + some accelerator very common (GPU, XEON Phi)
- heterogeneity on-a-chip (SoCs, APUs)
- cost efficiency dominated by energy-efficiency

today's large-scale FEM codes

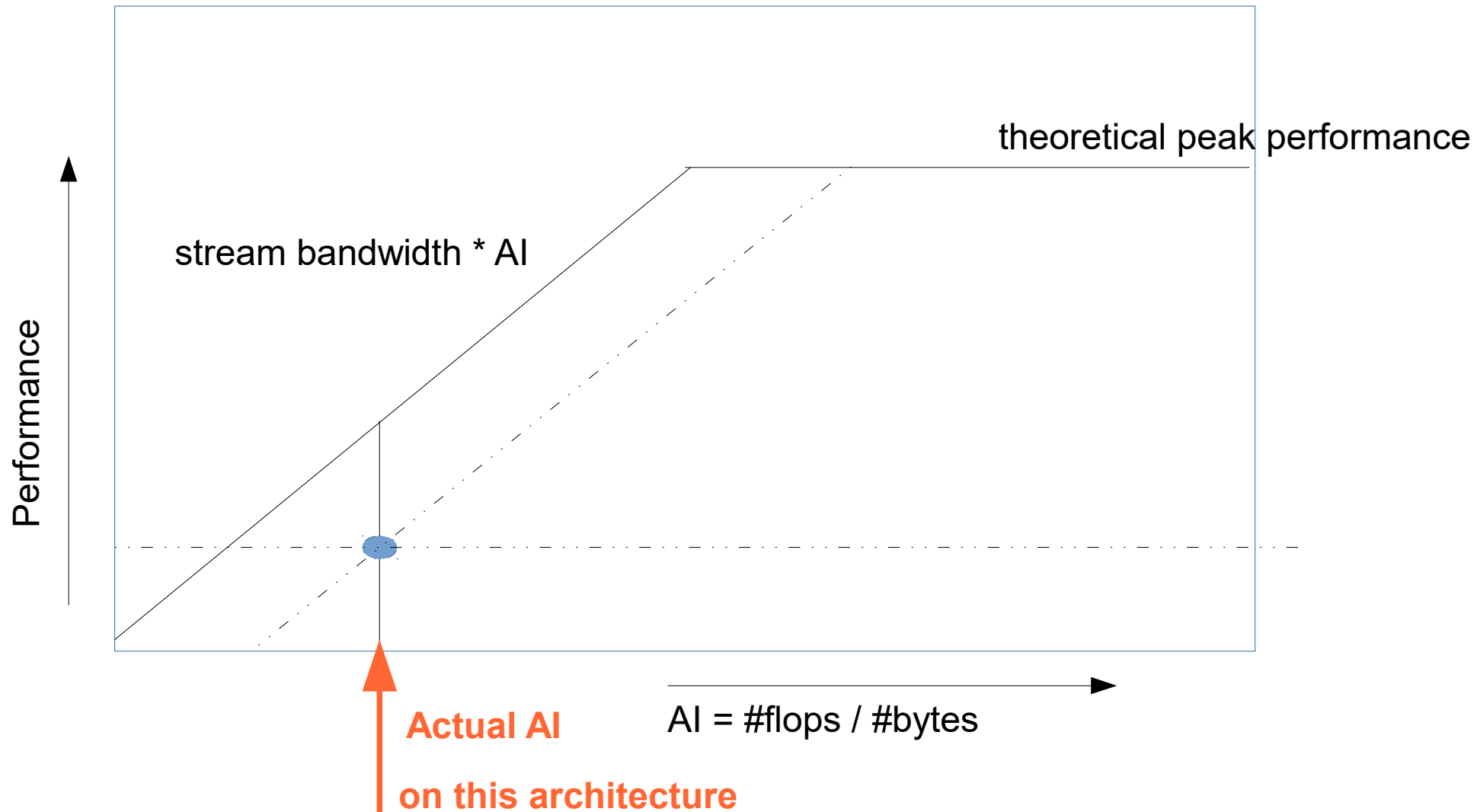
- have to adapt to target hardware
- heterogeneity and frameworking
- parallelisation of applications (DD mostly)
- parallelisation of core components (e.g. 'linear solver on GPU')
- optimisation with respect to many details (data flow and SIMD mostly)
- can we have the same results with less energy-consumption?

**Hardware evolution is usually out of
our control – software is not**

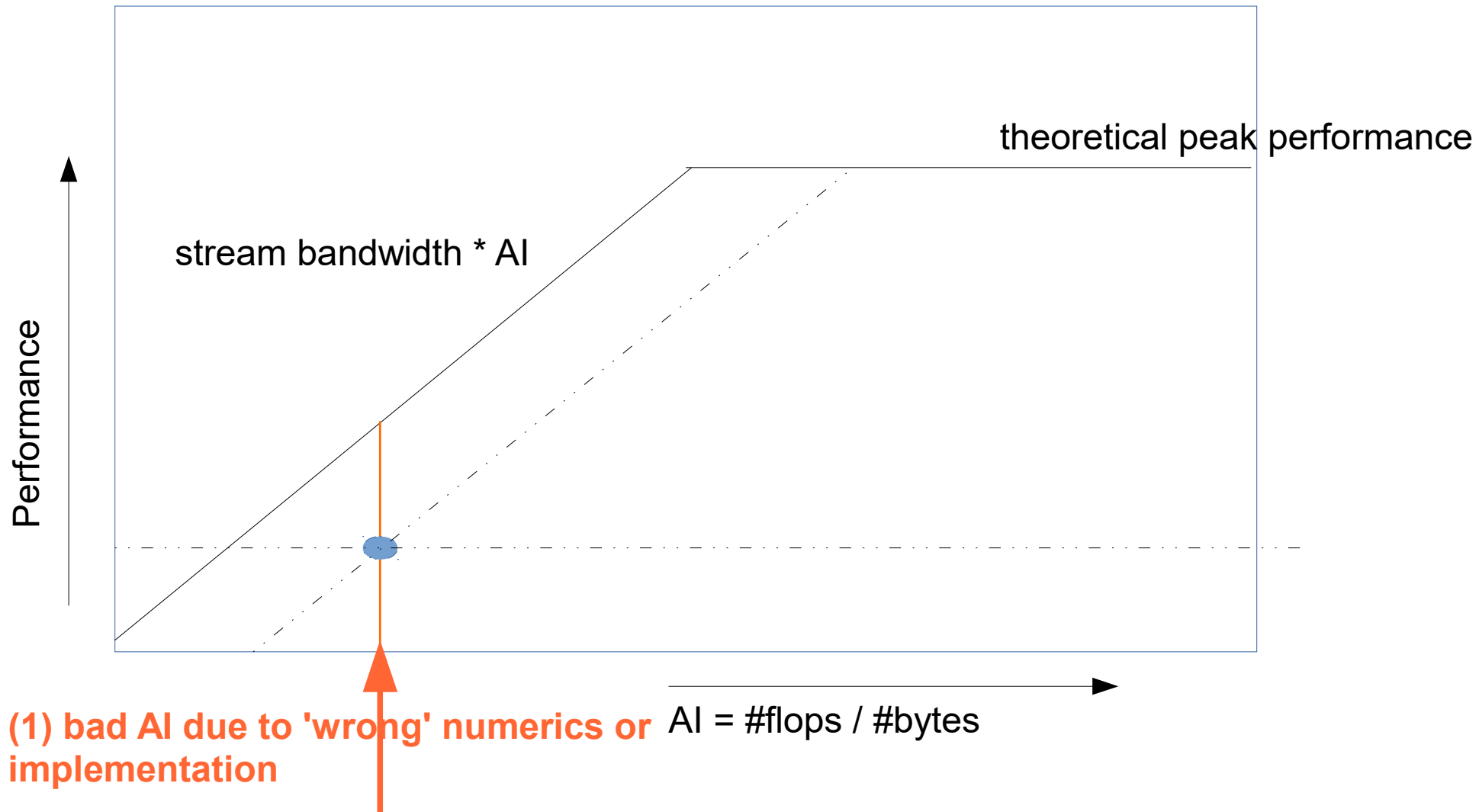
What we can expect from hardware



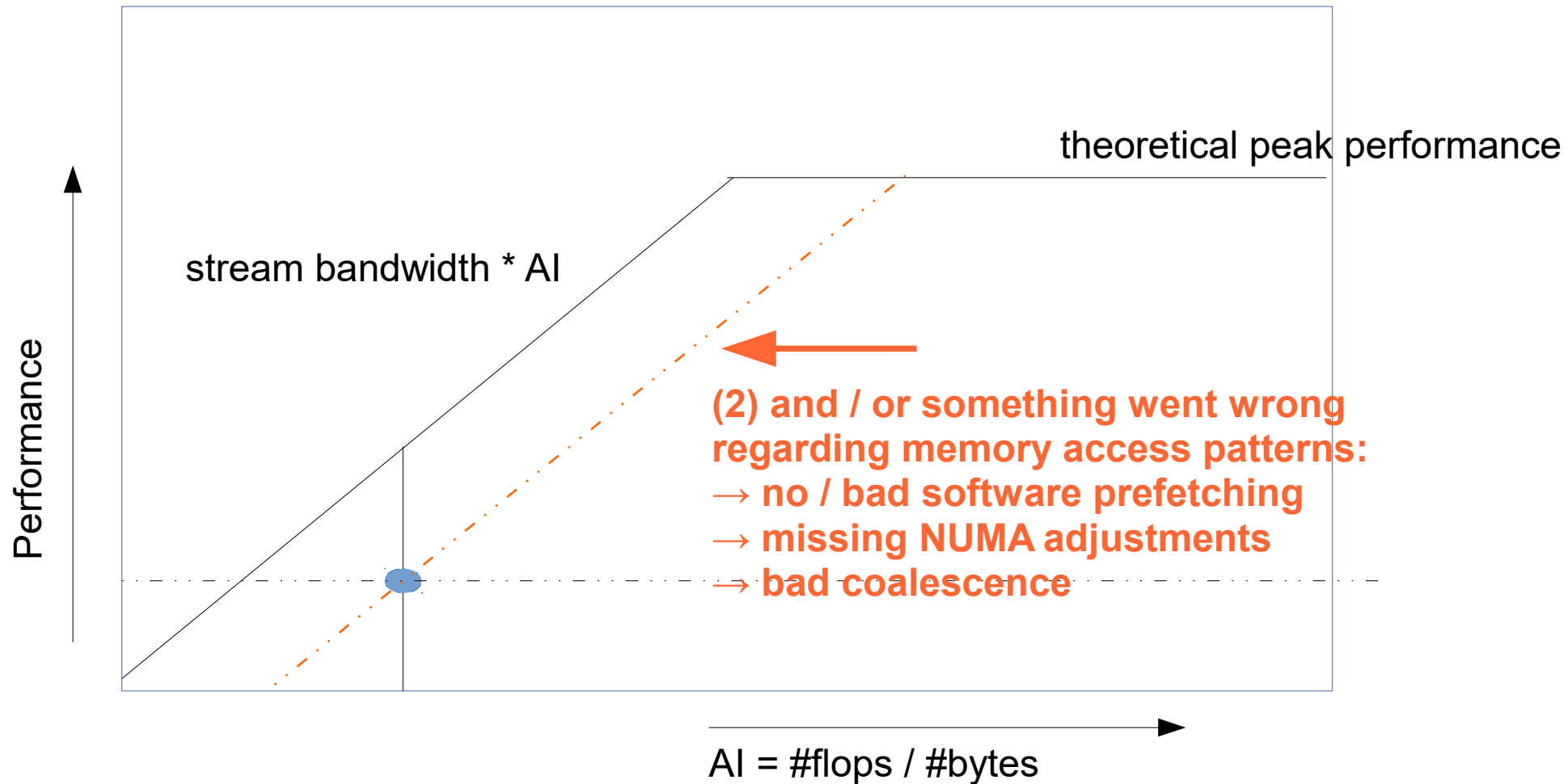
What we can expect from hardware



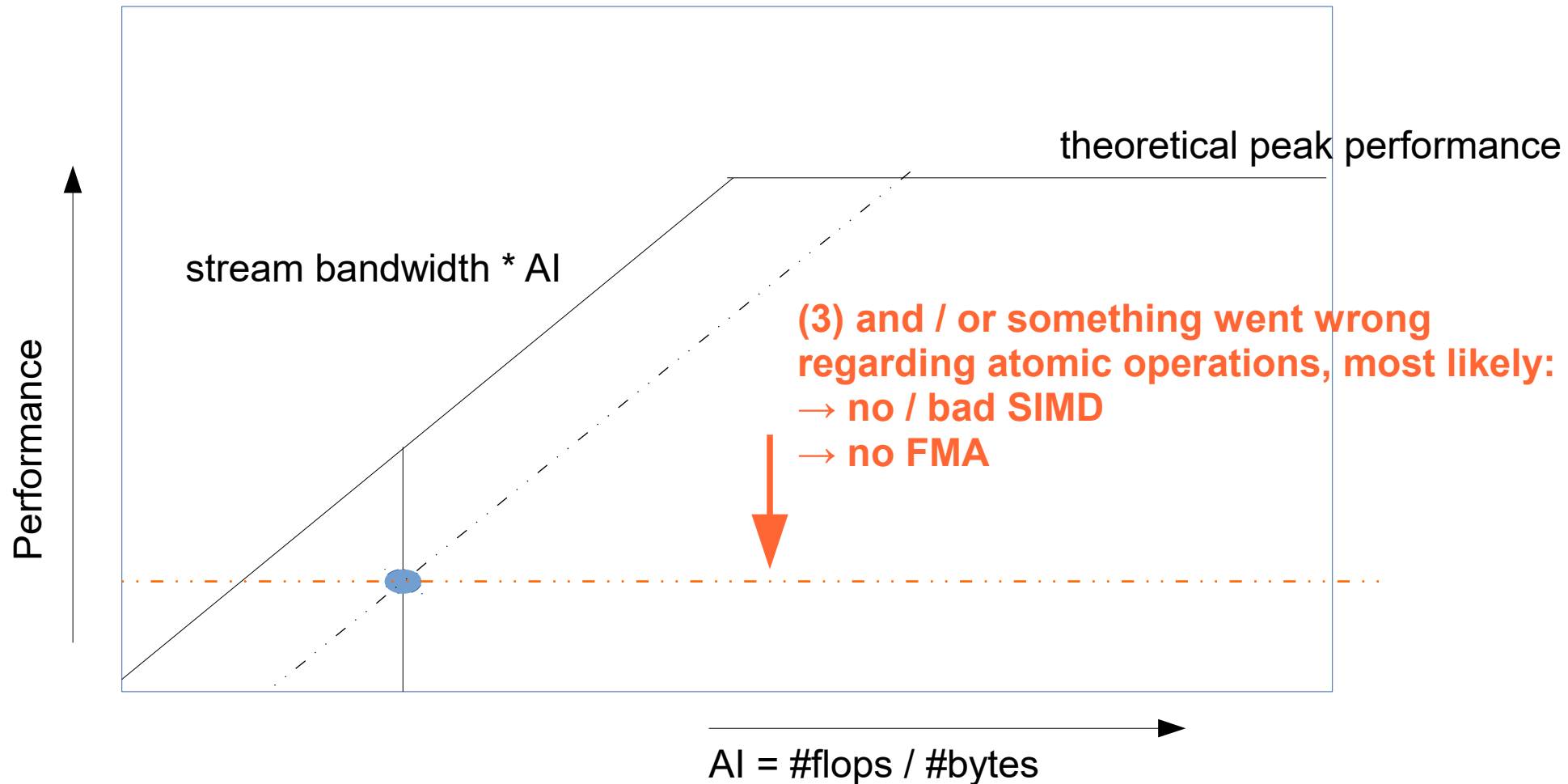
Three possibilities to tweak



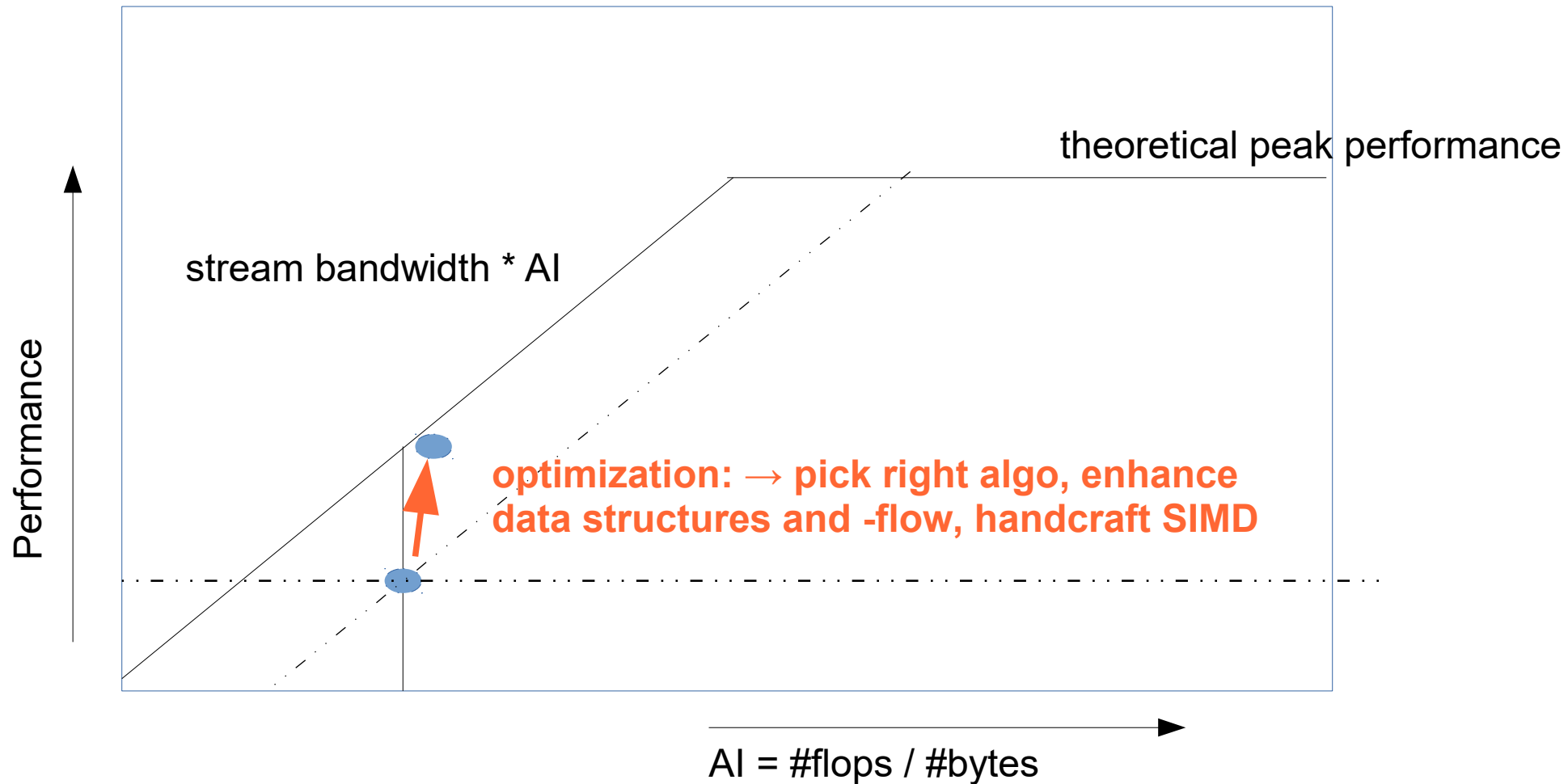
Three possibilities to tweak



Three possibilities to tweak

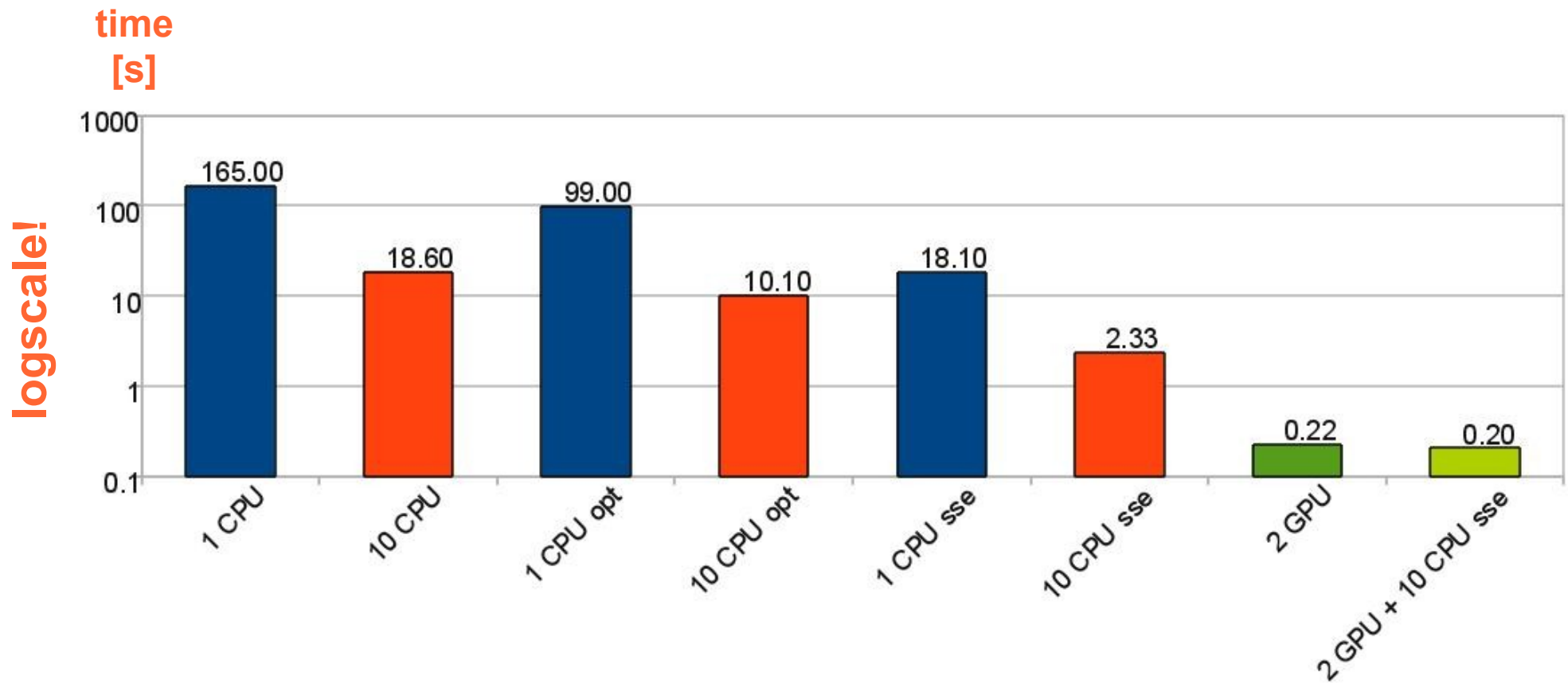


Three possibilities to tweak



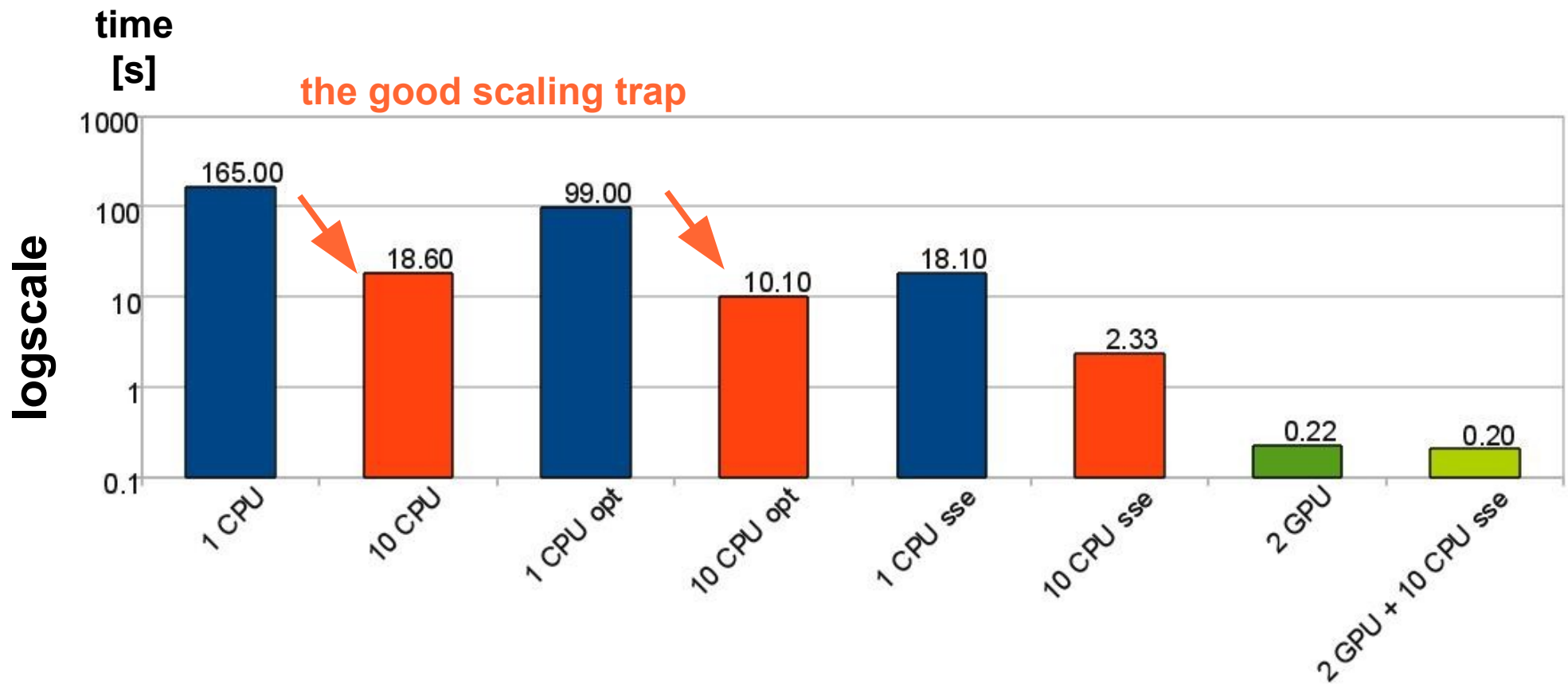
Hardware-oriented Numerics

(I) : Hardware Efficiency: apply 'classical' roofline models until optimal



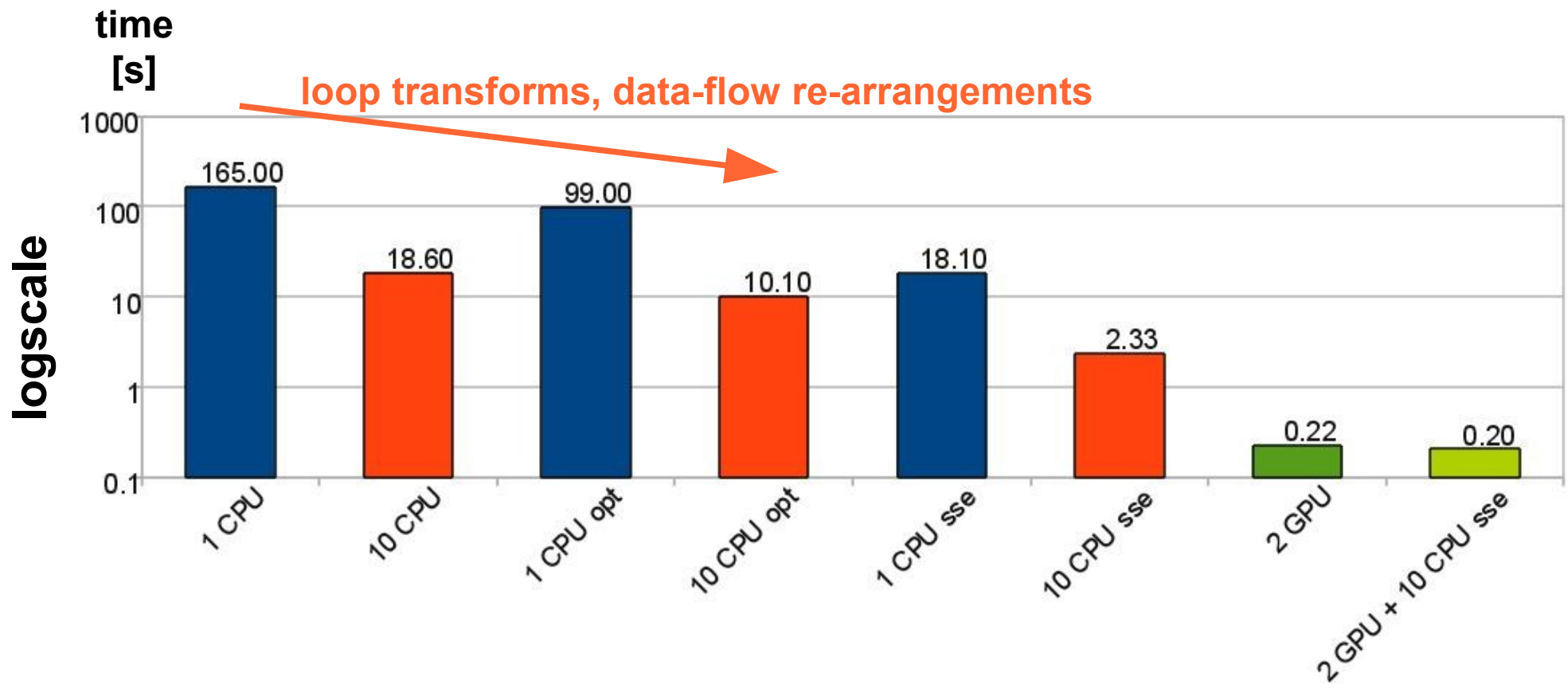
Hardware-oriented Numerics

(I) : Hardware Efficiency: apply 'classical' roofline models until optimal



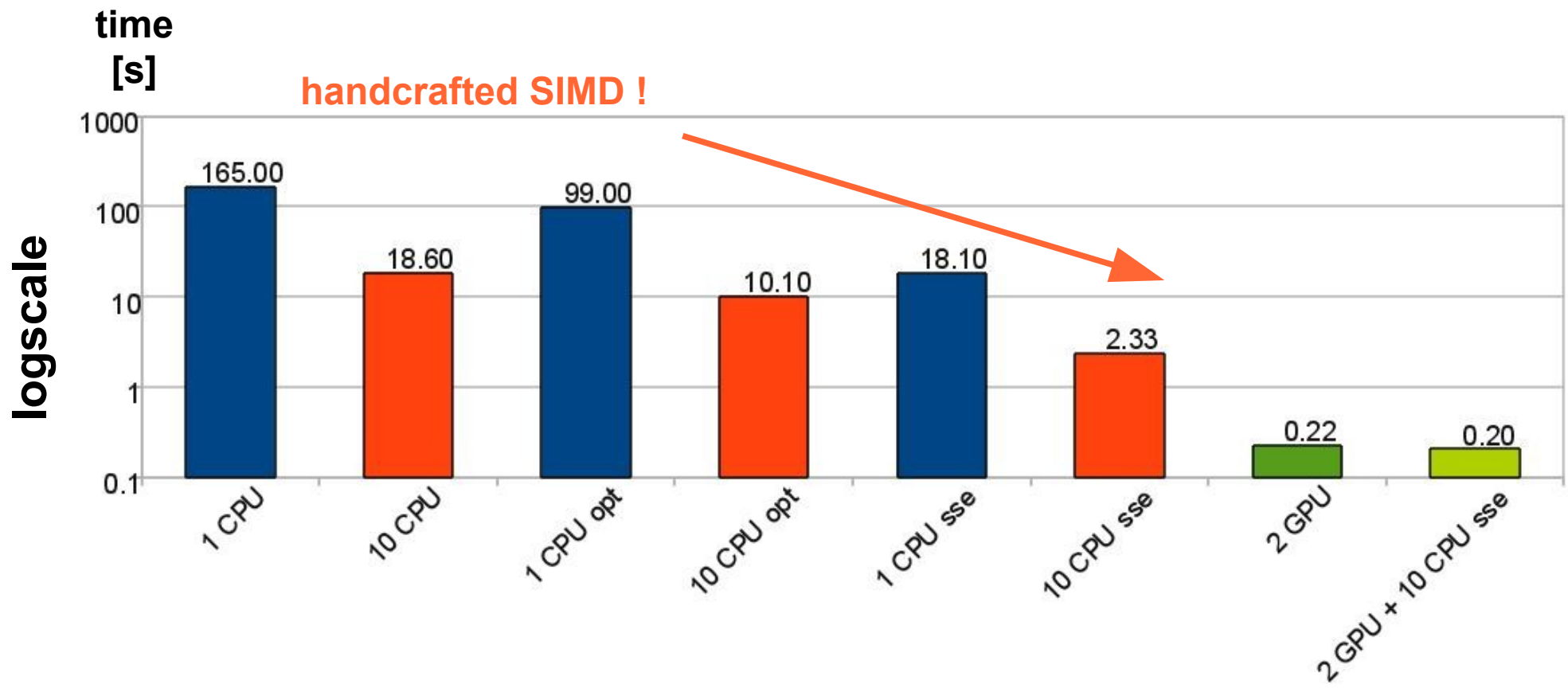
Hardware-oriented Numerics

(I) : Hardware Efficiency: apply 'classical' roofline models until optimal



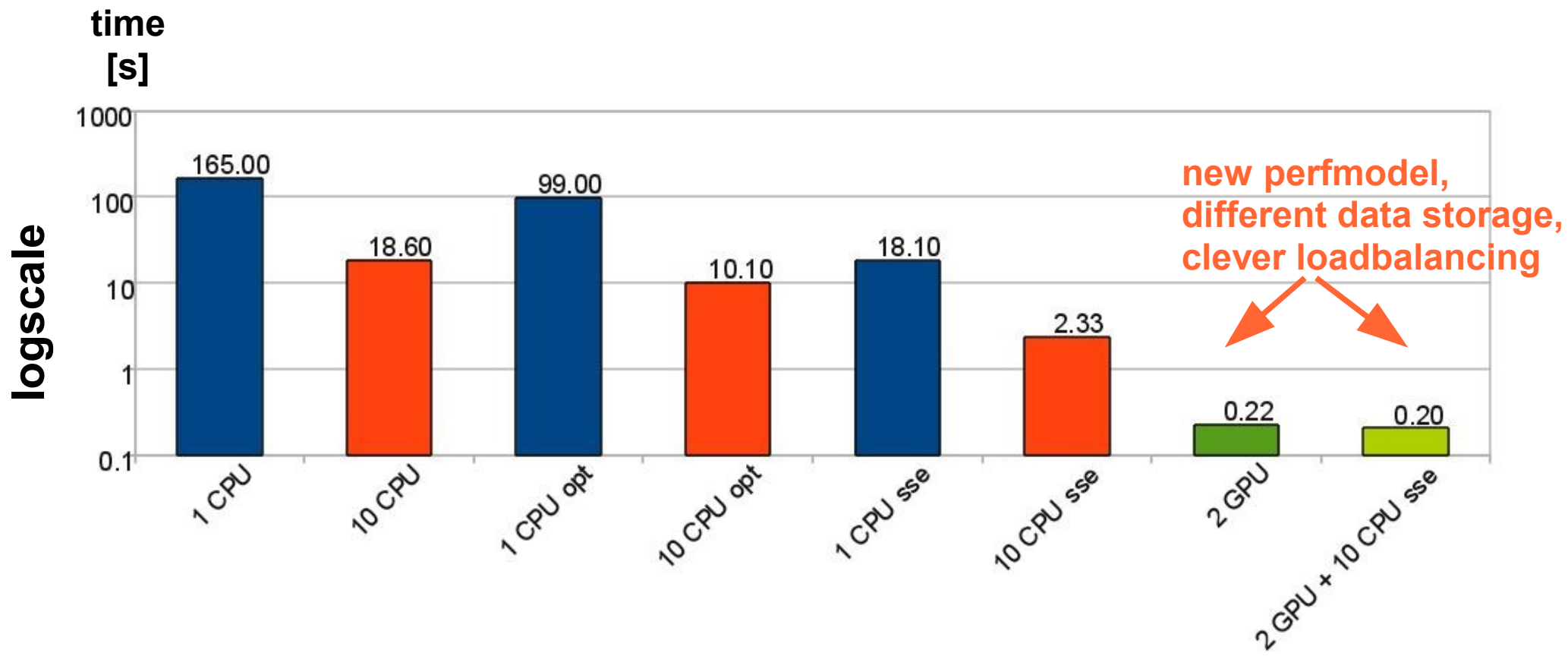
Hardware-oriented Numerics

(I) : Hardware Efficiency: apply 'classical' roofline models until optimal



Hardware-oriented Numerics

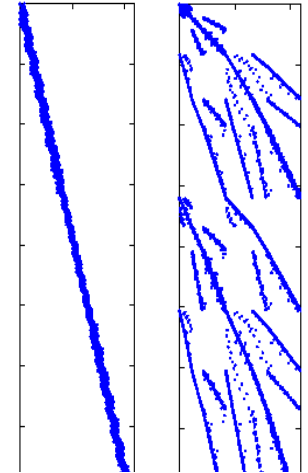
(I) : Hardware Efficiency: apply 'classical' roofline models until optimal



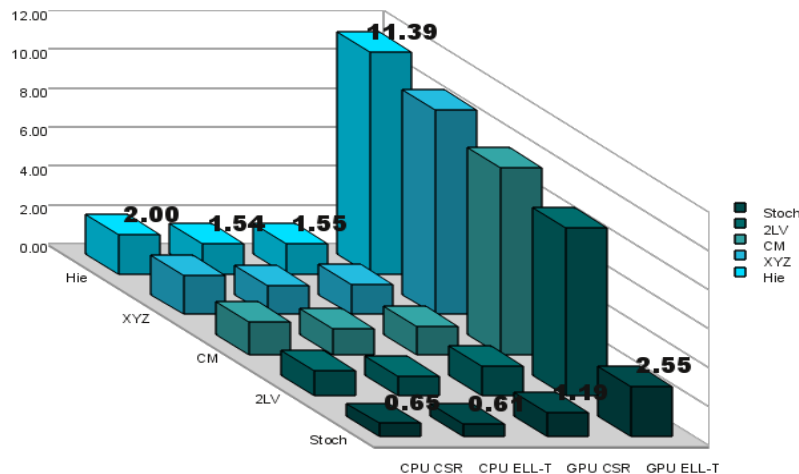
Hardware-oriented Numerics

(I) Hardware Efficiency: kernel-based optimisation: SpMV

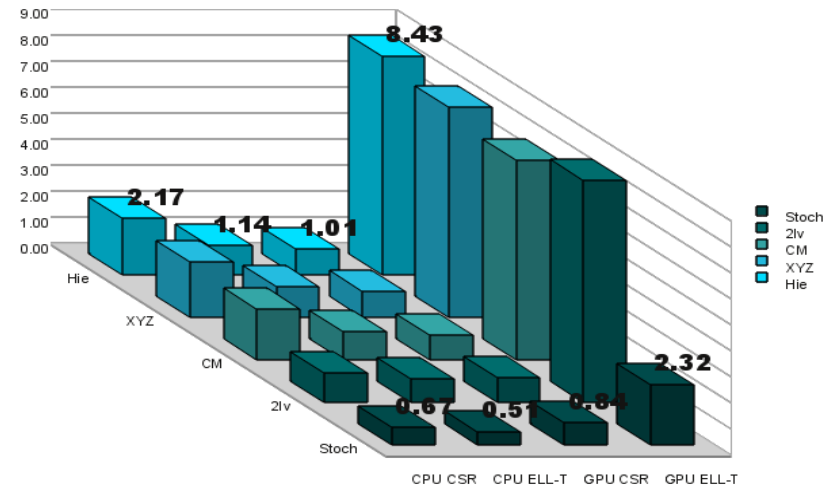
- one of the most prominent kernels in solving PDEs with high-end FEM
- memory access matters a lot
- hardware efficiency considerations start early: DOF numbering
- hardware-efficiency requires different matrix storage
- FE space matters



Performance
[Gflop/s]



Q1

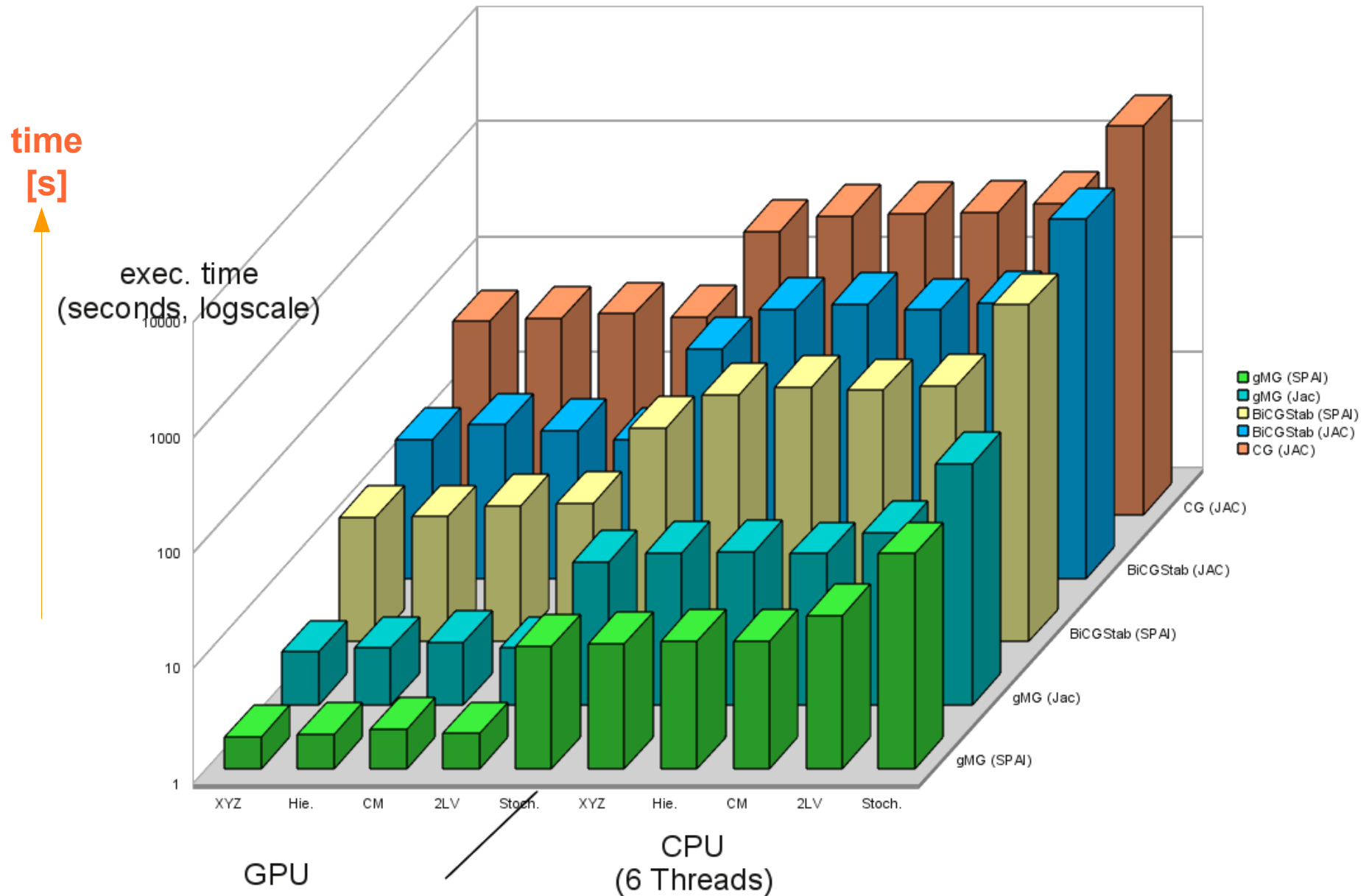


Q2

→ good, but: sole concentration on HE will not do the job

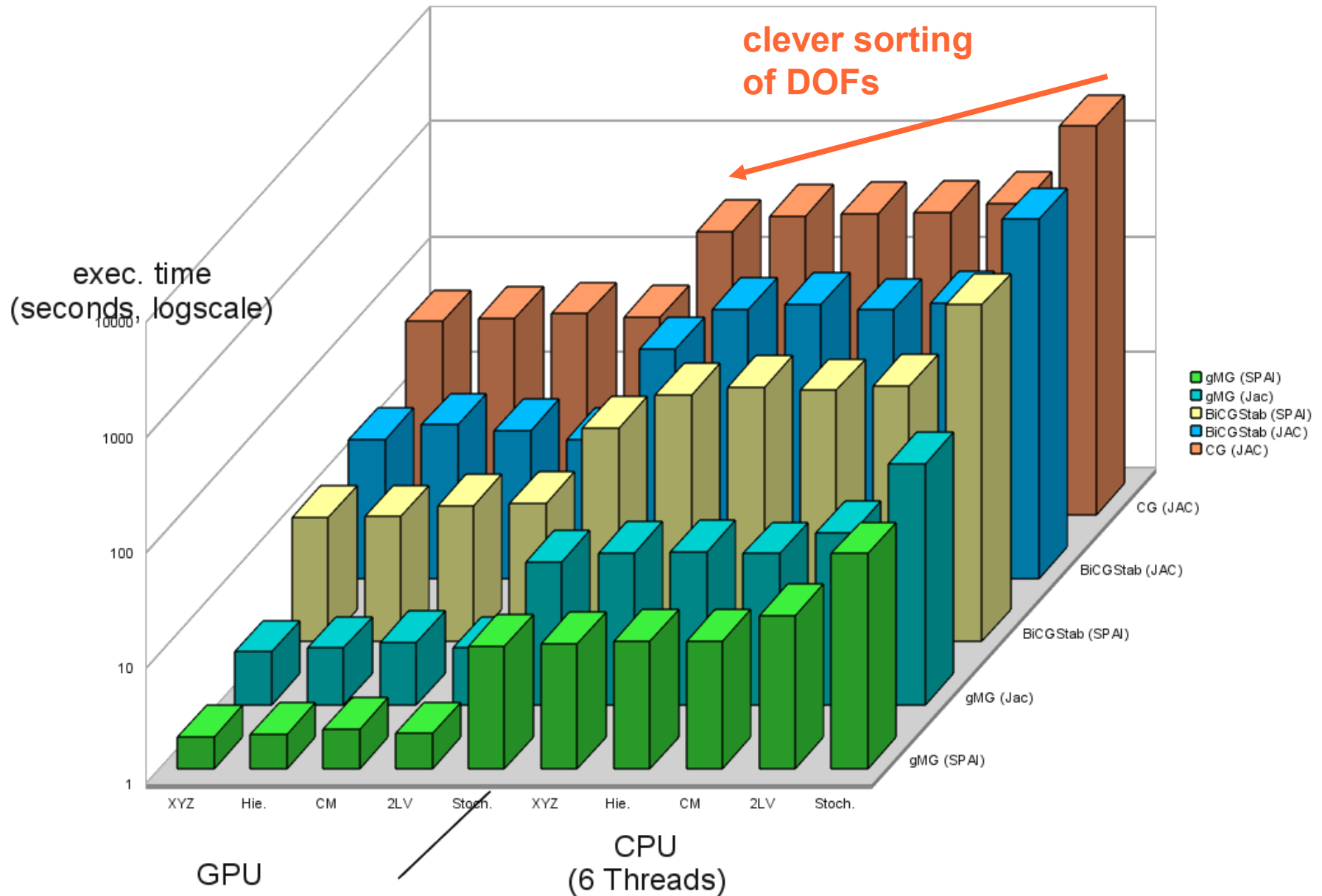
Hardware-oriented Numerics

(II) Numerical Efficiency



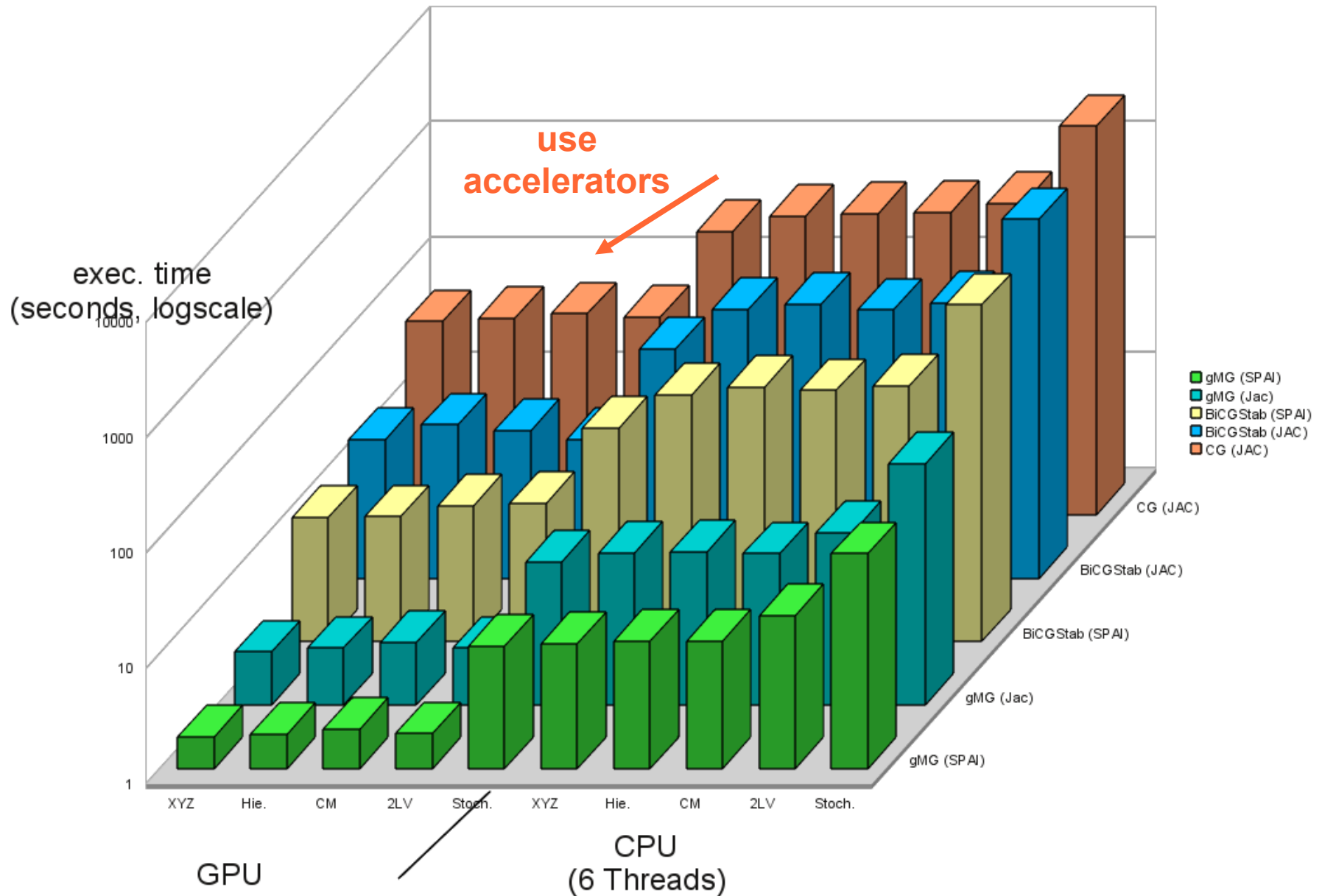
Hardware-oriented Numerics

(II) Numerical Efficiency



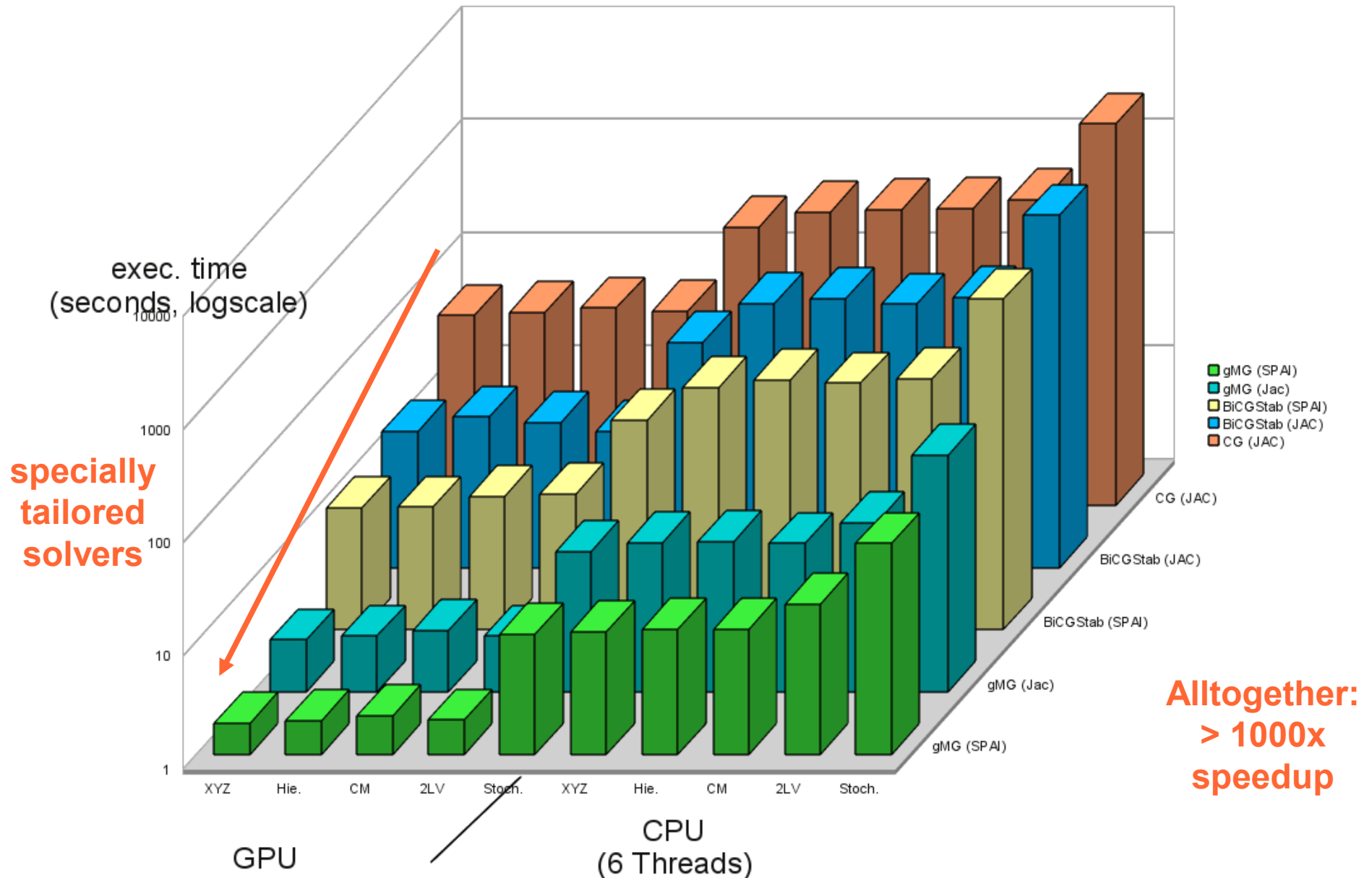
Hardware-oriented Numerics

(II) Numerical Efficiency



Hardware-oriented Numerics

(II) Numerical Efficiency

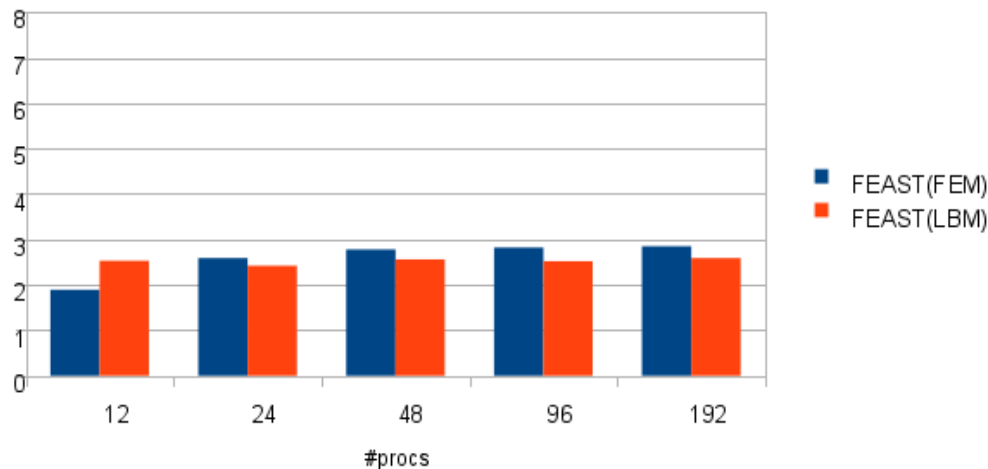


Hardware-oriented Numerics

(III) *Energy Efficiency* (?)

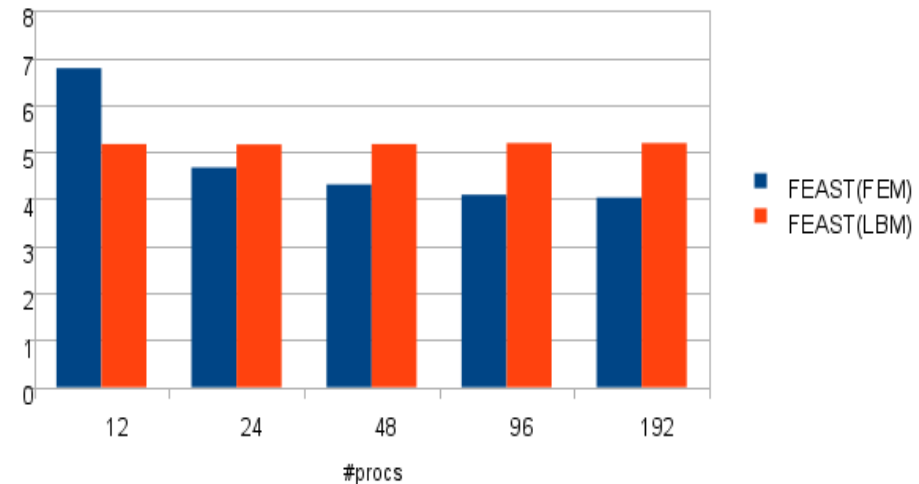
- energy consumption/efficiency is one of the major challenges for future supercomputers
→ exascale challenge
- we can not afford to go all 'macho-flops' any more
- in 2012 we proved: we can solve PDEs for less energy 'than normal'
- simply by switching computational hardware from commodity to embedded
- Tegra 2 (2x ARM Cortex A9) in the Tibidabo system of the MontBlanc project
- tradeoff between energy and wall clock time

energy down ARM vs x86



~3x less energy

speedup x86 vs ARM



~5x more time!

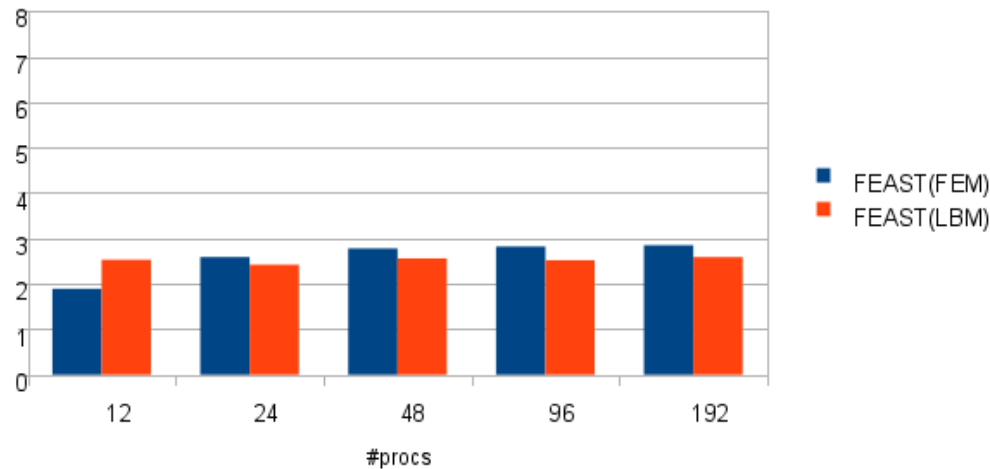
Hardware-oriented Numerics

(III) *Energy Efficiency* (?)

To be more energy efficient with different computational hardware, this hardware would have to dissipate *less power* at the *same performance* as the other!

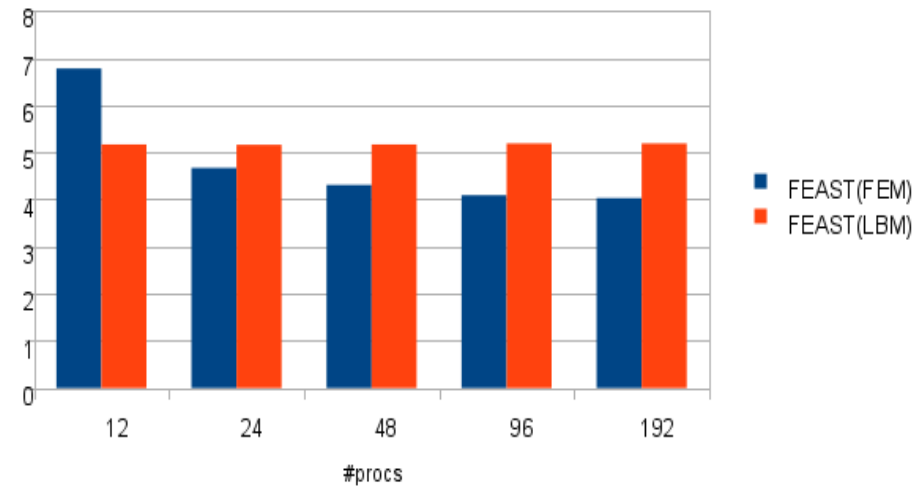
→ More performance per Watt!

energy down ARM vs x86



~3x less energy

speedup x86 vs ARM



~5x more time!

Hardware-oriented Numerics

(III) *Energy Efficiency*: technology of ARM-based SoCs since 2012

Something has been happening in the mobile-computing hardware evolution:

[one word in advance: there are many more SoC designs (like from TI, Qualcomm, ...)]

- Tegra 3 (late 2012) was also based on A9 but had 4 cores
- Tegra 4 (2013) is build upon the A15 core (higher frequency) and had more RAM and LPDDR3 instead of LPDDR2
- Tegra K1 (32 Bit, late 2014) CPU pretty much like Tegra 4 but higher freq., more memory

More importantly: TK1 went GPGPU and comprises a programmable Kepler GPU on the same SoC!

- the promise: 350+ Gflop/s for less than 11W
- for comparison: Tesla K40 + x86 CPU: 4200 Gflop/s for 385W
- 2.5x higher EE promised
- interesting for Scientific Computing! Higher EE than commodity!

Power consumption and performance of basic kernels

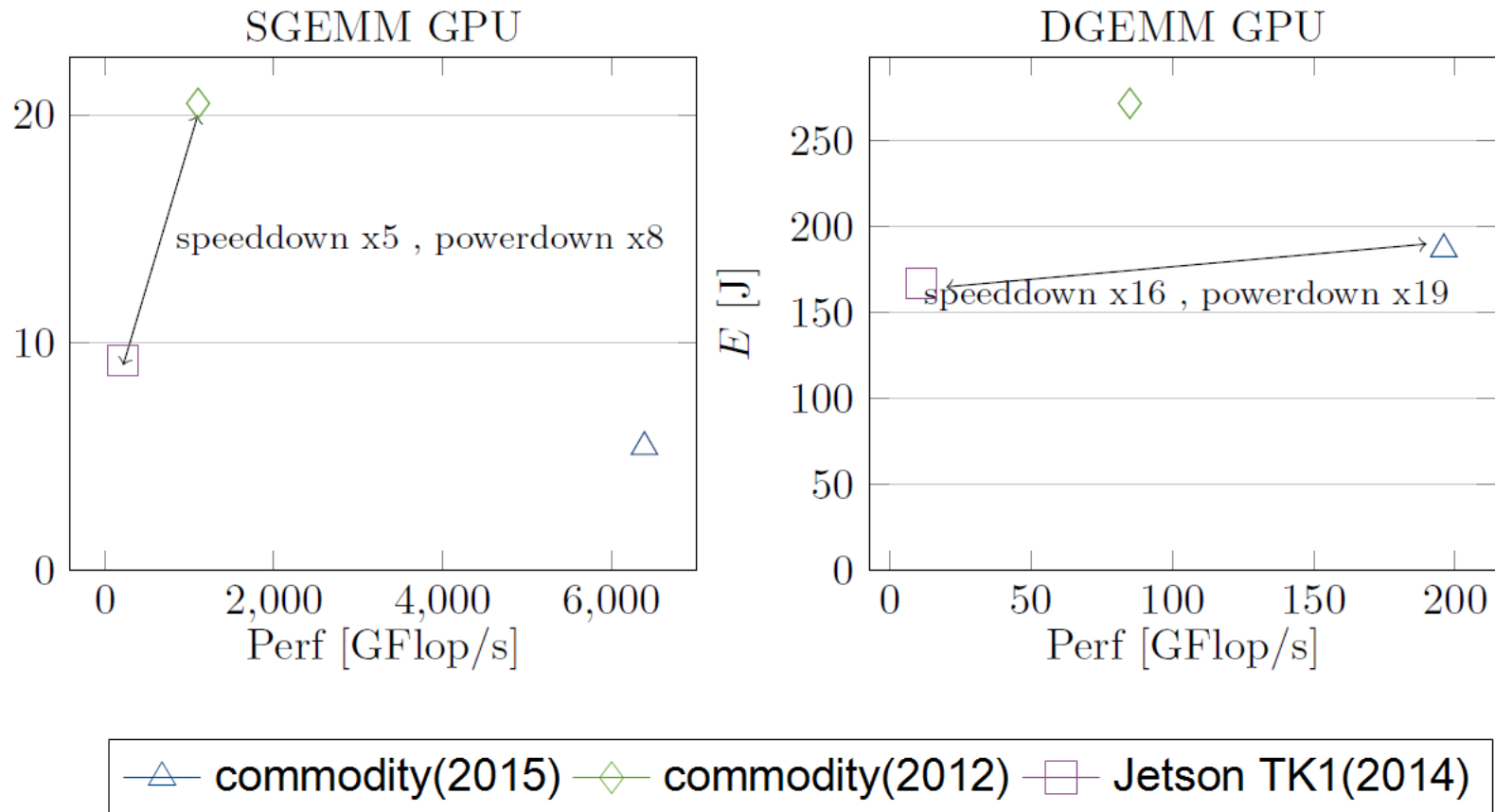
Testhardware

	i5-3470	i5-4690K	Jetson TK1
micro-architecture	Ivy Bridge	Haswell	Cortex-A15 (Tegra K1)
N_{cores}	4	4	4
clock speed	3.20 GHz (turbo 3.60 GHz)	3.50 GHz (turbo 3.9 GHz)	2.3 GHz
L1-cache	4x 32 KB + 4x 32 KB	4x 32 KB + 4x 32 KB	32 KB + 32 KB
L2- / L3-cache	4x 256 KB / 6 MB	4x 256 KB / 6 MB	2 MB / –
memory type	DDR3	DDR3	LPDDR3
peak memory bandwidth	25.6 GByte/s	25.6 GByte/s	14.9 GByte/s
P_{base}	51 W (Intel chipset)	41 W (Intel chipset)	3.9 W (Jetson TK1)
release date	Q2'12	Q2'14	Q2'14

	GeForce GTX 660 system	GeForce GTX 980 system	Jetson TK1
micro-architecture	GeForce GTX 660	GeForce GTX 980 Ti	GK20a (Kepler)
memory type	GDDR5	GDDR5	LPDDR3
peak memory bandwidth	144.2 GByte/s	336.5 GByte/s	14.9 GByte/s
peak performance (SP)	1881 GFlop/s	6054 GFlop/s	326 GFlop/s
peak performance (DP)	78 GFlop/s	189 GFlop/s	13 GFlop/s
P_{base}	41 W (Intel chipset)	51 W (Intel chipset)	3.9 W (Jetson TK1)
release date	Q3'12	Q2'15	Q2'14

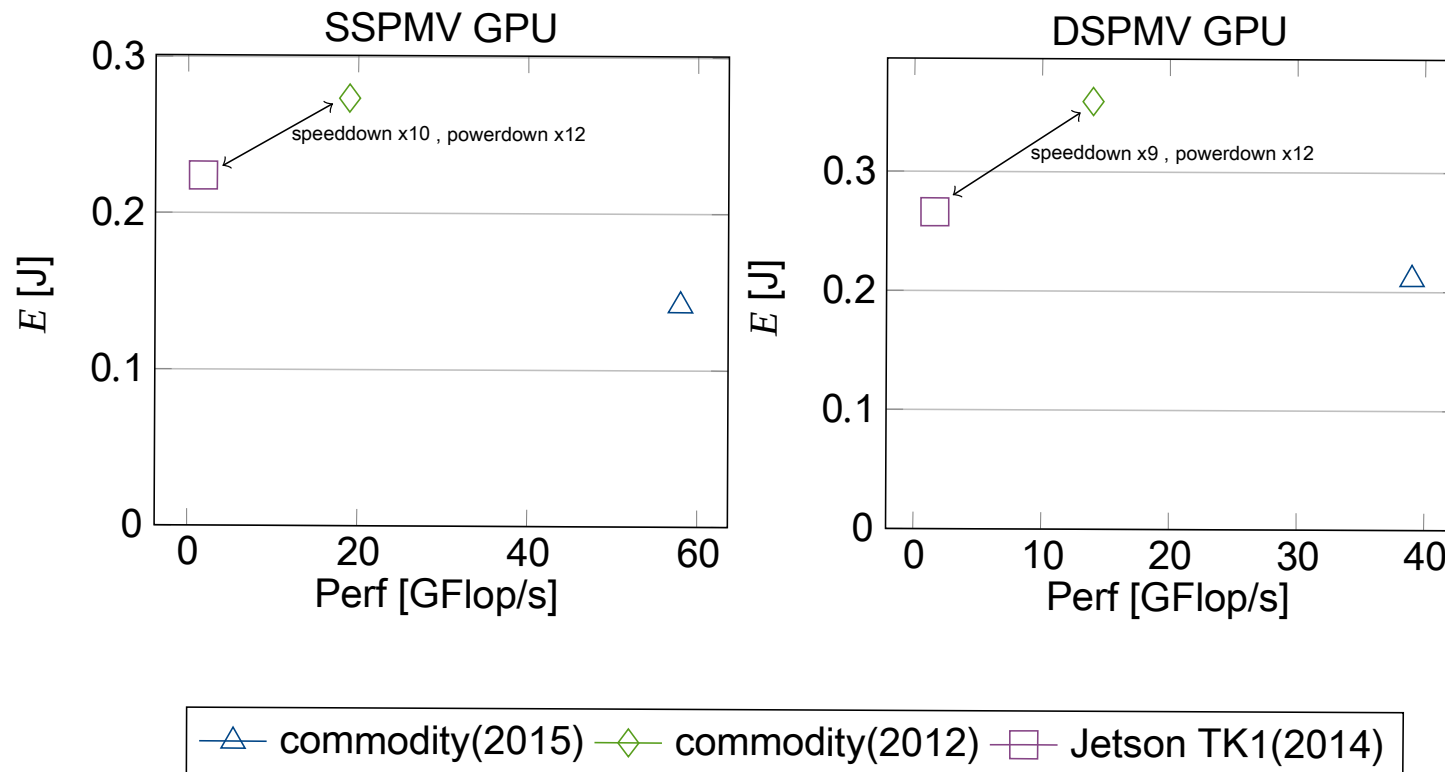
Power consumption and performance of basic kernels

S/DGEMM



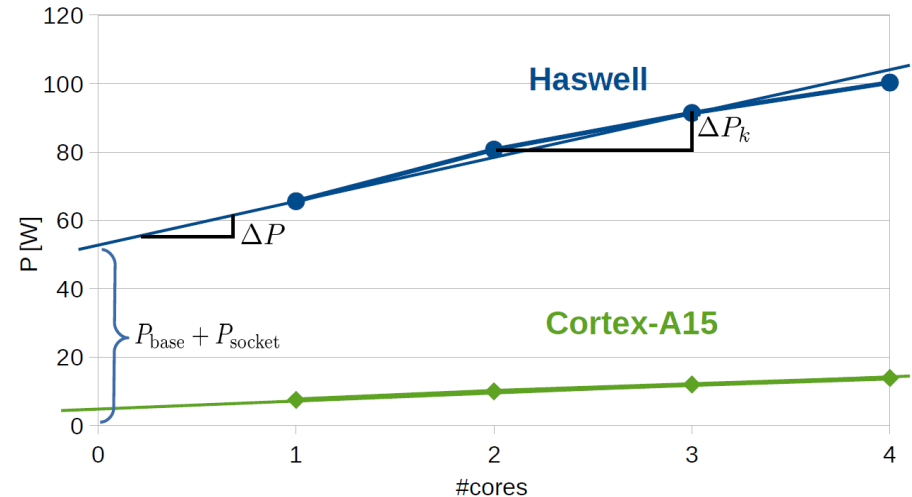
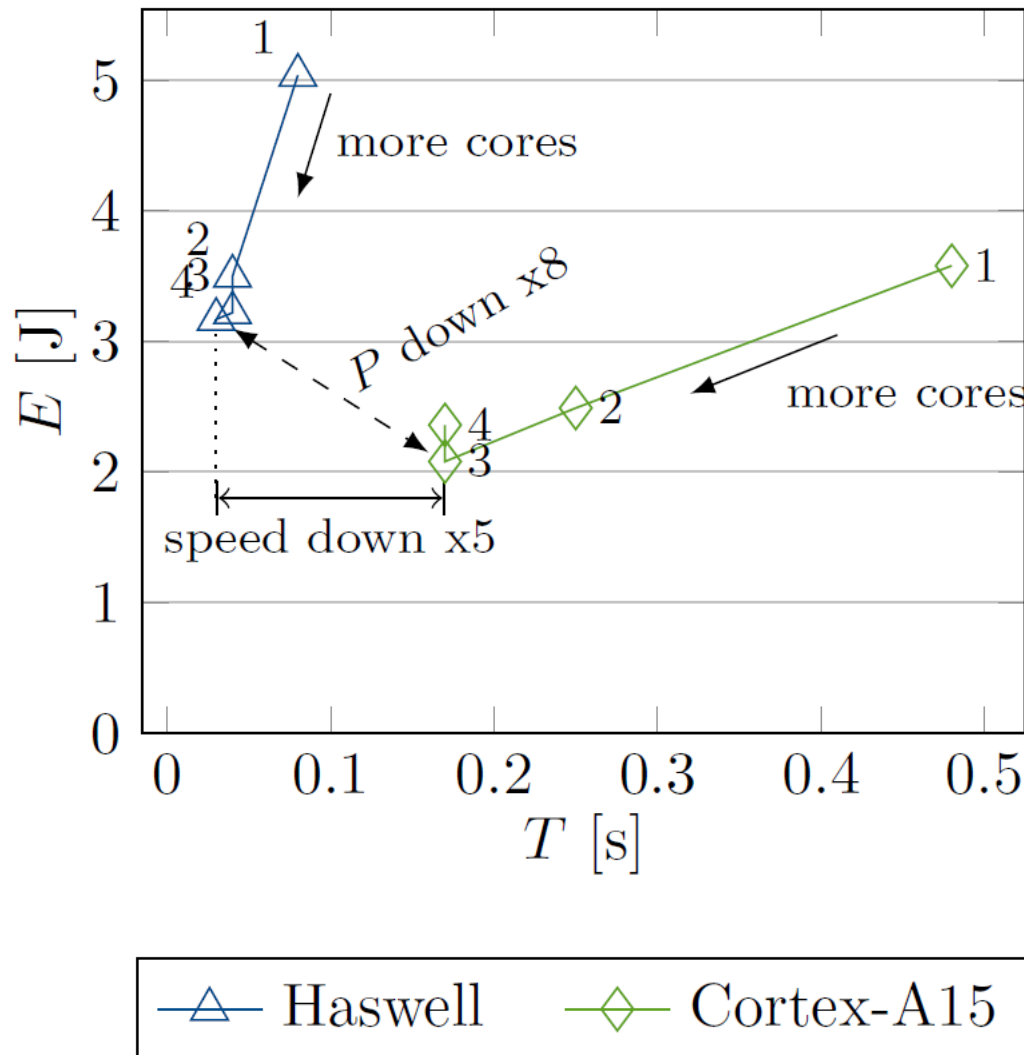
Power consumption and performance of basic kernels

S/DSPMV (ELL)



Applications

DG-FEM SWE solver (UTBEST3D, Erlangen)



$$\Delta T = \Delta T(k) = T_{k-1} - T_k, \quad \Delta P = \Delta P(k) = P_k - P_{k-1}$$

$$E(k) = P(k)T(k) = (P_{\text{base}} + P_{\text{socket}} + k\Delta P) T(k), \quad k \geq 1$$

- race-to-idle: cores are either 'on' or 'off' at preset frequency
- performance gain still positive with more threads
- at some #threads: disproportional performance-gain / power increase → **#threads = #cores not efficient**

Total efficiency?

(some) aspects of efficiency

- **numerical efficiency** dominates asymptotic behaviour and wall clock time
- **hardware-efficiency**
 - exploit all levels of parallelism provided by hardware (SIMD, multi-threading on a chip/device/socket, multi-processing in a cluster, hybrids)
 - then try to reach good scalability (communication optimisations, block comm/comp)
- **energy-efficiency**
 - by hardware:
 - what is the most energy-efficient computer hardware? What is the best core frequency? What is the optimal number of cores used?
 - by software as a direct result of performance

Hardware-oriented Numerics: Enhance hardware- and numerical efficiency simultaneously, use (most) energy-efficient Hardware(-settings) where available! Attention: codedependencies!

Today's major example: (local) unstructured grid geometric Multigrid with Approximate Inverse smoothers on GPUs

Example: smoother / preconditioner

weak smoother / preconditioner:

- e.g. Jacobi
- rapid execution on hardware
- large number of iterations in solver

strong(er) smoothing / preconditioning

- e.g. ILU
- hard to parallelize → slow execution on today's hardware
- small amount of iterations in solver

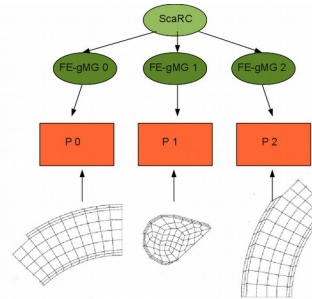
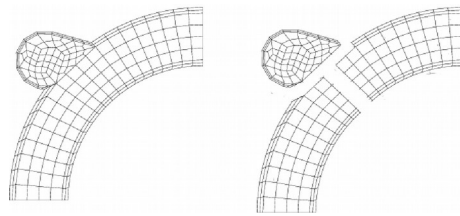
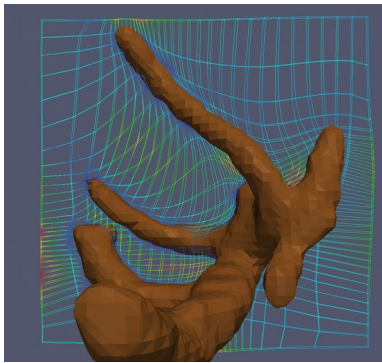
Which is better? What does better mean? Is there something in between?

One simple idea: base everything on SpMV...

...and pull the most out of it

FEgMG: context

- (local to a process) geometric multigrid for unstructured grids
- within a Schwarz-type domain decomposition: in global solver, use:



ScaRC-preconditioner:

- 1 $d \leftarrow b - Ax$
- 2 $y \leftarrow \sum_i \widetilde{R}_i^T \text{MG}(d)$, where $\text{MG}(d)$:
 - 1 $d_i \leftarrow R_i d$
 - 2 $y_i \leftarrow \text{FE-gMG}(B_i, d_i)$
 - 3 $y \leftarrow \sum_i \widetilde{R}_i^T y_i$
- 3 $x \leftarrow x + y$

FEgMG: ingredients

- (local) geometric multigrid for unstructured grids
- with Approximate Inverse smoothers
- with FE transfer operators
- with clever DOF sorting
- on GPUs (and multicore CPUs)
- all based on one kernel: SpMV

One simple idea: base everything on SpMV...

...and pull the most out of it

concentrate all tuning in one kernel: sparse matrix vector multiply (SpMV)

- in coarse-grid solver: preconditioned Krylov subspace methods
- smoother: preconditioned Richardson iteration or Krylov subspace method
- local preconditioners by approximate inverses
- grid transfers
- defect

the remainder

- a little BLAS1 (dot-product, norm, scale, ...)
- important: grid transfer operators can also be realised as SpMV

advantages

- flexibility (only matrices are switched): blackbox
- oblivious of FE-space, dimension, ...
- performance-tuning concentrated

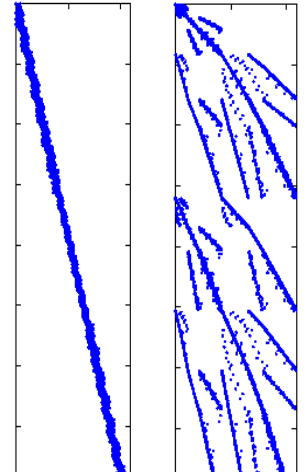
disadvantages

- we somewhat move the problem from solver to assembly of matrices

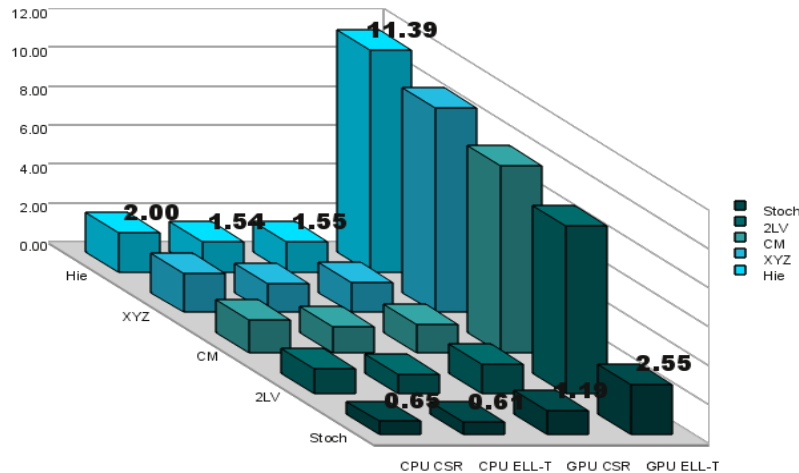
Hardware-oriented Numerics

(I) Hardware Efficiency: kernel-based optimisation: SpMV

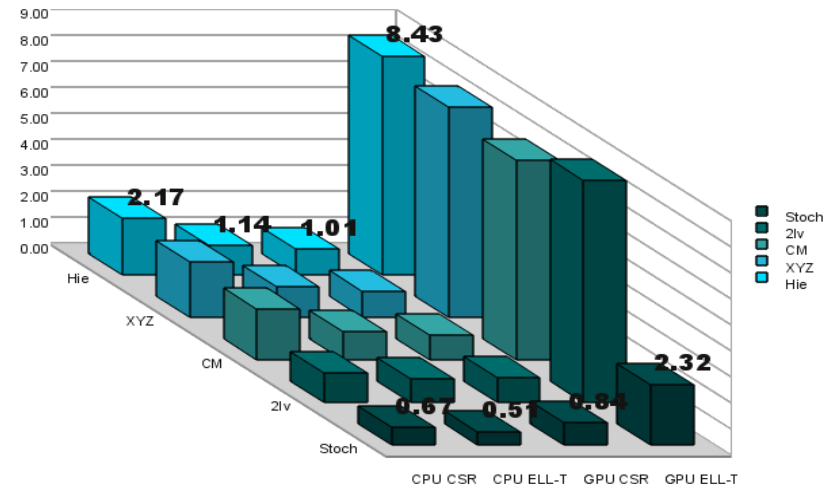
- one of the most prominent kernels in solving PDEs with high-end FEM
- memory access matters a lot
- hardware efficiency considerations start early: DOF numbering
- hardware-efficiency requires different matrix storage
- FE space matters



Performance
[Gflop/s]



Q1



Q2

→ good, but: sole concentration on HE will not do the job

SpMV in MG for GPU

ELLPACK-type matrix storage

- store sparse matrix S in two arrays A (non-zeros in column-major order) and j (column-index for each entry in A)
- A has ($\#$ rows in S) (maximum $\#$ non-zeros in rows of S)
- shorter rows are filled
- additional array rl to store effective non-zeros count per row (get stop on row right)

advantages

- complete regular access pattern to y and A
- GPU implementation:
 - one thread for each element y_i
 - access to all ELLPACK-R arrays and y completely coalesced (column-major)
- access on x : use texture-cache (FERMI and up : L2-cache)
- no synchronisation between threads needed
- no branch-divergence
- in addition: multiple threads can access one row (ELLPACK-T)
- access to x depends on non-zero pattern of A ! bandwidth given by → DOF-numbering

$$S = \begin{bmatrix} 1 & 7 & 0 & 0 \\ 0 & 2 & 8 & 0 \\ 5 & 0 & 3 & 9 \\ 0 & 6 & 0 & 4 \end{bmatrix} \Rightarrow A = \begin{bmatrix} 1 & 7 & * \\ 2 & 8 & * \\ 5 & 3 & 9 \\ 6 & 4 & * \end{bmatrix} \quad j = \begin{bmatrix} 0 & 1 & * \\ 1 & 2 & * \\ 0 & 2 & 3 \\ 1 & 3 & * \end{bmatrix} \quad rl = \begin{bmatrix} 2 \\ 2 \\ 3 \\ 2 \end{bmatrix}$$

smoother: SPAI

SPAI

$$\| I - M_{\text{SPAI}} A \|_F^2 = \sum_{k=1}^n \| \mathbf{e}_k^T - \mathbf{m}_k^T A \|_2^2 = \sum_{k=1}^n \| A^T \mathbf{m}_k - \mathbf{e}_k \|_2^2$$

→ where \mathbf{e}_k is the k -th unit vector and the k -th row of $M = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n]$:

→ for n columns of M → n least squares opt-problems:

$$\min_{\mathbf{m}_k} \| A^T \mathbf{m}_k - \mathbf{e}_k \|_2, \quad k = 1, \dots, n.$$

→ use non-zero pattern of the stiffness matrix for M

→ put it into a Richardson Iteration

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \omega M_{\text{SPAI}} (\mathbf{b} - A \mathbf{x}^k)$$

FegMG: SPAI(1) as a smoother

Q1	CPU					GPU						
	Jacobi		SPAI			Jacobi			SPAI			
sort	time	#iter	time	#iter	speedup jac	time	#iter	speedup cpu	time	#iter	speedup jac	speedup cpu
2lv	4.04	13	2.54	5	1.59	1.06	13	3.82	0.56	5	1.88	4.53
CM	3.65	13	2.19	5	1.66	1.03	13	3.55	0.72	5	1.43	3.05
XYZ	3.48	13	2.06	5	1.69	0.98	13	3.53	0.51	5	1.93	4.04
Stoch	4.04	13	2.57	5	1.57	1.74	13	2.33	1.04	5	1.66	2.46
Hie	3.49	13	2.07	5	1.69	0.97	13	3.59	0.50	5	1.94	4.14

Q2	CPU					GPU						
	Jacobi		SPAI			Jacobi			SPAI			
sort	time	#iter	time	#iter	speedup jac	time	#iter	speedup cpu	time	#iter	speedup jac	speedup cpu
2lv	13.19	22	4.87	5	2.71	2.27	22	5.80	0.93	5	2.44	5.22
CM	11.40	22	4.40	5	2.59	2.50	22	4.56	1.02	5	2.44	4.30
XYZ	11.29	22	4.21	5	2.68	2.41	22	4.69	0.99	5	2.44	4.26
Stoch	12.92	22	5.14	5	2.51	4.78	22	2.70	2.04	5	2.35	2.52
Hie	11.25	22	4.24	5	2.66	2.44	22	4.60	1.00	5	2.43	4.22

FEAT → DUNE: first steps

use **SPAI in DUNE**: in a BiCGStab iteration, use as **preconditioner**:

→ Jacobi iteration

$$S_{\text{JAC}}^{\omega} : \mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \omega M_{\text{JAC}}(\mathbf{b} - A\mathbf{x}^k), k = 1, \dots, K$$

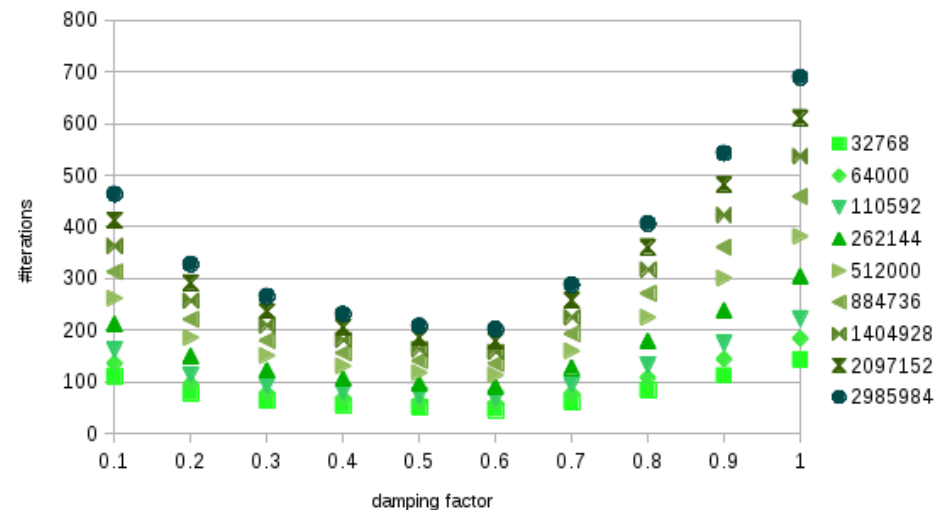
→ DG-block Jacobi iteration

$$S_{\text{BJAC}} : \mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + M_{\text{BJAC}}(\mathbf{b} - A\mathbf{x}^k), k = 1, \dots, K$$

$$M_{\text{BJAC}} = \sum_i R_i^T A_i^{-1} R_i$$

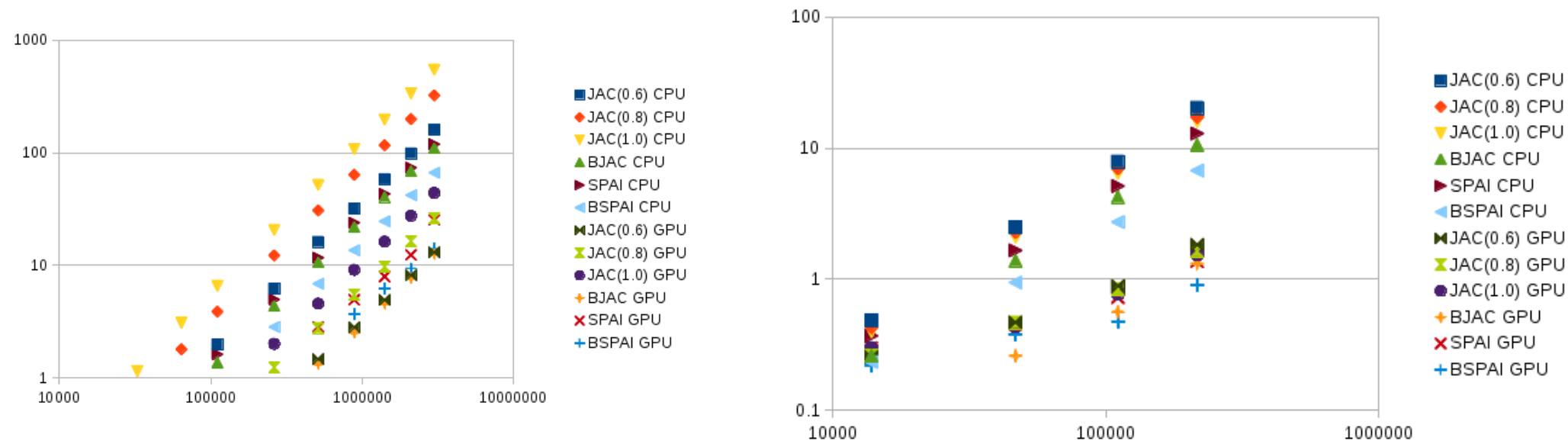
→ SPAI direct application

$$S_{\text{SPAI}} : \mathbf{x} \leftarrow M_{\text{SPAI}} \mathbf{x}$$



FEAT → DUNE

First results: DG1 and DG2



HPC centers

Current supercomputers

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science [AICS] Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945
6	Swiss National Supercomputing Centre [CSCS] Switzerland	Piz Daint - Cray XC30, Xeon E5-2670 8C 2.600GHz, Aries interconnect , NVIDIA K20x Cray Inc.	115,984	6,271.0	7,788.9	2,325
7	King Abdullah University of Science and Technology Saudi Arabia	Shaheen II - Cray XC40, Xeon E5-2698v3 16C 2.3GHz, Aries interconnect Cray Inc.	196,608	5,537.0	7,235.2	2,834
8	Texas Advanced Computing Center/Univ. of Texas United States	Stampede - PowerEdge C8220, Xeon E5-2680 8C 2.700GHz, Infiniband FDR, Intel Xeon Phi SE10P Dell	462,462	5,168.1	8,520.1	4,510
9	Forschungszentrum Juelich [FZJ] Germany	JUQUEEN - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	458,752	5,008.9	5,872.0	2,301
10	DOE/NNSA/LLNL United States	Vulcan - BlueGene/Q, Power BQC 16C 1.600GHz, Custom Interconnect IBM	393,216	4,293.3	5,033.2	1,972

Mflop/s / W

1902

2143

2177

830

2 – 17 MW of power!

Simulation comes at huge cost!

<http://www.top500.org/lists/2015/06/>

Green HPC

Greenmost supercomputers are 'unconventional'

	Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
Japan	1	7,031.58	RIKEN	Shoubu - ExaScaler-1.4 80Brick, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SC	50.32
Japan	2	6,842.31	High Energy Accelerator Research Organization /KEK	Suiren Blue - ExaScaler-1.4 16Brick, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband PEZY-SC	28.25
Japan	3	6,217.04	High Energy Accelerator Research Organization /KEK	Suiren - ExaScaler 32U256SC Cluster, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, PEZY-SC	32.59
Germany	4	5,271.81	GSI Helmholtz Center	ASUS ESC4000 FDR/C3S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150	57.15
Japan	5	4,257.88	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x	39.83
USA	6	4,112.11	Stanford Research Computing Center	XStream - Cray CS-Storm, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR, Nvidia K80	190.00
USA	7	3,962.73	Cray Inc.	Storm1 - Cray CS-Storm, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, Nvidia K40m	44.54
USA	8	3,631.70	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	52.62
Germany	9	3,614.71	TU Dresden, ZIH	Taurus GPUs - Bull bullx R400, Xeon E5-2680v3 12C 2.5GHz, Infiniband FDR, Nvidia K80	58.01
USA	10	3,543.32	Financial Institution	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	54.60

Accelerators rule the field, unconventional design is leading.

Green HPC

Greenmost supercomputers are 'unconventional'

Top500 rank	Green500 Rank	MFLOPS/W	Site*	Computer*	Total Power (kW)
160	1	7,031.58	RIKEN	Shoubu - ExaScaler-1.4 80Brick, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband FDR, PEZY-SC	50.32
392	2	6,842.31	High Energy Accelerator Research Organization /KEK	Suiren Blue - ExaScaler-1.4 16Brick, Xeon E5-2618Lv3 8C 2.3GHz, Infiniband, PEZY-SC	28.25
366	3	6,217.04	High Energy Accelerator Research Organization /KEK	Suiren - ExaScaler 32U256SC Cluster, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, PEZY-SC	32.59
215	4	5,271.81	GSI Helmholtz Center	ASUS ESC4000 FDR/G2S, Intel Xeon E5-2690v2 10C 3GHz, Infiniband FDR, AMD FirePro S9150	57.15
	5	4,257.88	GSIC Center, Tokyo Institute of Technology	TSUBAME-KFC - LX 1U-4GPU/104Re-1G Cluster, Intel Xeon E5-2620v2 6C 2.100GHz, Infiniband FDR, NVIDIA K20x	39.83
	6	4,112.11	Stanford Research Computing Center	XStream - Cray CS-Storm, Intel Xeon E5-2680v2 10C 2.8GHz, Infiniband FDR, Nvidia K80	190.00
	7	3,962.73	Cray Inc.	Storm1 - Cray CS-Storm, Intel Xeon E5-2660v2 10C 2.2GHz, Infiniband FDR, Nvidia K40m	44.54
	8	3,631.70	Cambridge University	Wilkes - Dell T620 Cluster, Intel Xeon E5-2630v2 6C 2.600GHz, Infiniband FDR, NVIDIA K20	52.62
	9	3,614.71	TU Dresden, ZIH	Taurus GPUs - Bull bullx R400, Xeon E5-2680v3 12C 2.5GHz, Infiniband FDR, Nvidia K80	58.01
	10	3,543.32	Financial Institution	iDataPlex DX360M4, Intel Xeon E5-2680v2 10C 2.800GHz, Infiniband, NVIDIA K20x	54.60

Still not developed under the premise of EE, power source not included in thinking yet

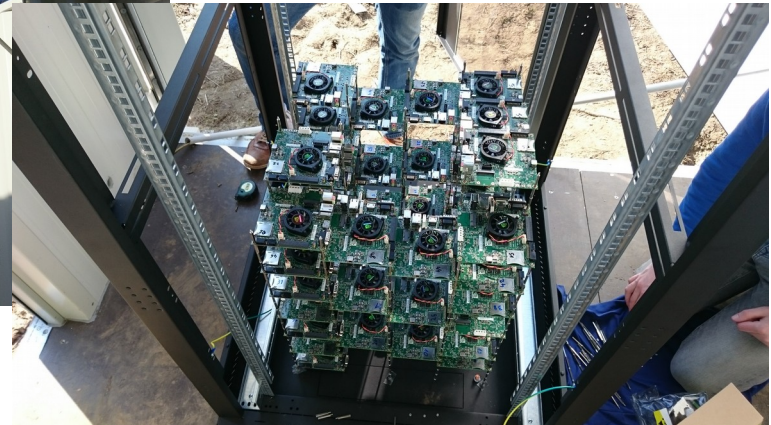
Unconventional HPC for EE

Bring together the two pillars of Energiewende for HPC

- Renewable power source
- Energy Efficiency

→ **Design the hardware for EE**

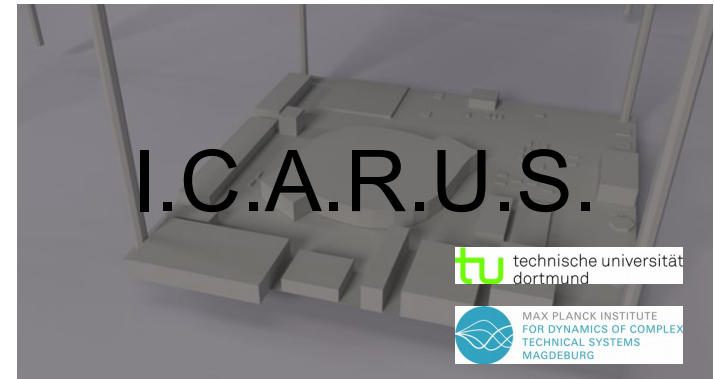
→ **Design the software for the hardware by using HWON**



A compute center of the future (?)

Vision

- Insular
- Compute-center for
- Applied Mathematics with
- Renewables-provided power supply based on
- Unconventional compute hardware empaired with
- Simulation Software for Technical Processes



Motivation

- **system integration** for Scientific HPC
 - high-end unconventional compute hardware
 - high-end renewable power source (photo-voltaic)
 - **specialy tailored numerics and simulation software: high end Mathematics**
- **no future spendings due to energy consumption**
- **SME-class resource: <80K€**
- **Scalability, modular design**
- (simplicity)
- (maintainability)
- (safety)
- ...

I.C.A.R.U.S.

Whitesheet

- **nodes:** 60 x NVIDIA Jetson TK 1
- **#cores** (ARM Cortex-A15): 240
- **#GPUs** (Kepler, 192 cores): 60
- **RAM/core:** 2GB LPDDR3
- **switches** (GiBit Ethernet): 3xL1, 1xL2
- **cluster theoretical peak perf:** ~20TFlop/s SP
- **cluster peak power (including cooling/heating):** < 2kW, provided by PV

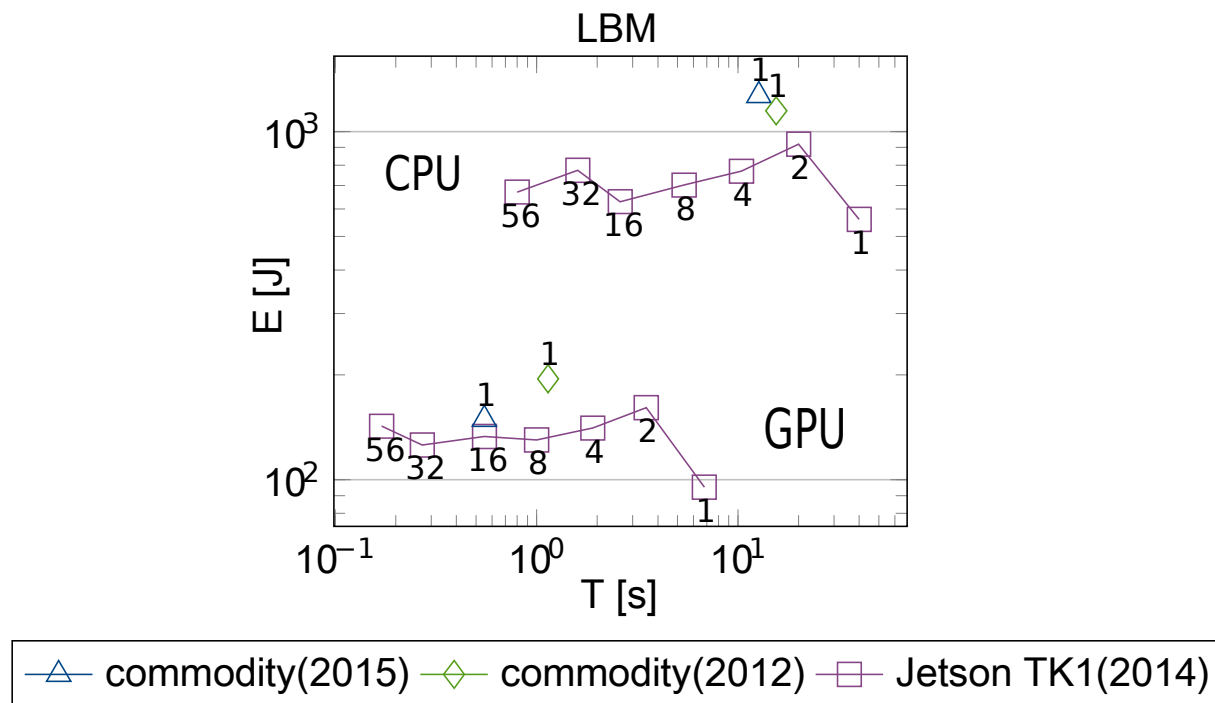
- **storage:** 10+1 BananaPI Boards comprising:
 - 1 TB Western Digital Eco HDD
 - 2 Dual Core ARM (1 GHz, 1 GB RAM)
 - GigabitEthernet networking
 - SATA
 - plus 16 GB eMMC internal (OS) and 128 GB SD swap / scratch per node

- **Software:** FEAT (optimised for Tegra K1): www.featflow.de

Applications

LBM SWE solver

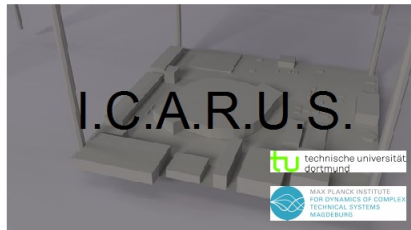
- complex things happen in apps
 - frequent kernel-switches
 - **many frequency-adjustments?**



Conclusion

- HWON is threefold now: EE comes into play
- → smaller power dissipation alone is not the deal
→ performance modelling/-engineering of software for EE is needed
- Hardware-/Software Co-Design can be a starting point:
 - Embedded tech has a different history than commodity hardware
 - Energy Efficiency is just starting to arrive in HPC
 - System Integration with state-of-the-art PV tech (or other renewables) is promising
- The I.C.A.R.U.S. computer and its housing/energy-source plus the FEAT software together offers a valuable ressource aiming at SMEs/University departments

Thank you



www.icarus-green-hpc.org

This work was also supported (in part) by the German Research Foundation (DFG)
through the Priority Programme
1648 'Software for Exascale Computing' (grant TU 102/48).

I.C.A.R.U.S. hardware is financed by MIWF NRW under the lead of MERCUR.