Energy efficiency aspects of high performance computing for PDEs

Dominik Göddeke (and the FEAST group)

Institut für Angewandte Mathematik (LS3)
TU Dortmund
dominik.goeddeke@math.tu-dortmund.de

25th Biennial Numerical Analysis Conference
University of Strathclyde, Glasgow, UK
June 25, 2013

technische universität dortmund
fakultät für mathematik

Hardware (r)evolution

Parallelism, specialisation and heterogeneity

- Frequency scaling is over, we now scale 'cores' and SIMD width
- Visible on all architectural levels
  - Fine-grained: SSE/AVX, GPU 'threads'
  - Medium-grained: GPUs, MICs, CPUs, NUMA within a node
  - Coarse-grained: MPI between heterogeneous nodes
- Even CPU-only systems are heterogeneous
- These are fun days for programmers

This paradigm shift is here to stay

- Power is the root cause of this
- Goal of this talk: try to explain why, and present a fun experiment with novel architectures

Hardware background
(and some politics)

Energy consumption of big machines

Distribution between various components (source: DOE)

- 40–60 %: processors and memories
- 10 %: interconnect and storage
- Remainder (up to 50 %!): infrastructure, i.e. lighting, cooling, PSU waste, backup power supplies, ...

Three principal attack vectors towards better energy efficiency

- More energy-efficient nodes
  - Much wider fine-grained parallelism
  - Low-power architectures not (initially) designed for HPC
  - Implication: reduced cooling cost, e.g. passive instead of active heatsinks
- More efficient cooling: improved airflow, hot-water cooling, ...
- 'Recycling' of dissipated heat

Energy consumption of big machines

Current technology

- Top500 #1 (June 2013): Xeon + Xeon Phi
  - 33.9 PFLOP @ 17.8 MW: 1.9 GFLOP/W
- Green500 #1 (Nov. 2012): Xeon + Xeon Phi
  - 112 TFLOP @ 44.9 kW: 2.5 GFLOP/W

Extrapolation to exascale

- Top500 machine: $(1000/33.9) \cdot 17.8 = 525$ MW
- Green500 machine: 400 MW

Consequences

- Consensus: power envelope of ≈ 20 MW 'feasible'
- Need (at least) 20–25 times more energy efficient technology
- Where will this factor come from?

High-level view

Short digression into electrical engineering

- Power is proportional to voltage^2 $\times$ frequency
- Frequency is proportional to voltage (at same process size)
- Similarly bad: power also proportional to temperature

A simple back-of-the-envelope calculation

	cores	V	freq	perf $\sim f$	power $\sim V^3$	power eff. perf/power
singlecore	1	1	1	1	1	1
faster singlecore	1	1.5	1.5	1.5	3.3	0.45
dualcore	2	0.75	0.75	1.5	0.8	1.88

A-ha!

- 50% more performance with 20% less power
- That's why everything has shifted to multi- and manycore

Low-level view: Dally's example

A typical yet hypothetical contemporary processor

- 64-bit FPU: 50 pJ for a multiply-add operation, 0.1 mm^2 of die area
- Moving one DP word through a channel of length 1 mm: 25 pJ
- Chip with 4000 FPUs (area 20 mm^2): 500 pJ to move one DP word across the chip, e.g., from a pin to an FPU
- Moving the same data off-chip: 1 nJ

Moving data is expensive in terms of energy

- Factor of 20 for this FPU (60 for two inputs and one output)
- My personal explanation why Xeon Phi doesn't scale in hardware

Same holds even more for instructions (von Neumann)

- Solution: SIMD, amortise instruction cost (fetch, decode, operand decode, ...) over several data items

My controversial slide

Ignoring SIMD in computations

- Conventional CPUs: 2–8× penalty (128–256 bit SSE/AVX registers, i.e. 2 doubles to 8 floats per instruction)
- GPUs: 16–64× penalty (32 32 bit values in NVIDIA warps, 64 32 bit values in AMD wavefronts)
- Xeon Phi: 8–16× penalty (512 bit registers)

Memory accesses are parallel in hardware since i386 (1985!)

- Cache line granularity of 64 byte on current CPUs
  - Access one float (4 byte) in a memory segment of 64 byte
  - 64 byte transferred
  - Never use the other 15 floats: 1/16 of peak memory performance
- NVIDIA GPUs: Cache lines of 32 byte and 128 byte
- Plus: computation is *much* faster than memory access

The 'money wall' problem

A simple rule of thumb

- 1 MW/a = 1 M EUR/a
  - Depends on your deal with the local energy company

Real numbers: our machine in Dortmund

- Deployed in 2009 (#249 in the TOP500) for roughly 1 M EUR
- 440 dual-quadcore nodes, 50 TFLOP/s, designed for capacity
- Roughly 85 % average load per year
- Annual electricity bill 220 K EUR (plus 60 K EUR for A/C)
- Accumulated running cost now exceeds initial acquisition cost

Who covers these costs?

- Funding agencies certainly do not

Scientific computing on low-power architectures

Promising 'new' processor architecture

Smartphones and tablets

- Tremendous market volume
- Probably a couple hundred GFLOP/s in this room
- Powerful enough for HD video decoding, mobile gaming, ...

Under the hood

- Low-power processor designs stemming from embedded systems now capable enough for HPC: dedicated FPUs, multicores, vector registers, reasonably-sized pipelines ...
- Almost all these chips contain IP by ARM Ltd., e.g. Apple A4/A5, ARM Cortex-A series, Samsung, ...
- Batteries matter, so very energy-efficient

Can we use these chips to solve real problems?

What is 'energy efficiency'?

In fact, a tricky question for application people

- Flops/Watt not meaningful at application level
- Slower units more energy-efficient even if it takes longer?
- Many more units to reach same speed still more energy-efficient?
- Time-to-solution and energy-to-solution!

Weak and strong scaling equally critical

- Weak: need many more units because of much lower memory/node
- Strong: need many more units to compensate for slower per-node execution
- Application-specific sweet spots, how much slower can we afford to be if it cuts our electricity bill in half?

Prototype cluster: tibidabo @ BSC

Machine details

- 96 dual-core NVIDIA Tegra-2 SoCs based on ARM Cortex-A9
- Hosted in SECO Q7 carrier boards (nodes, essentially developer kits)
- 1 GB memory per dualcore, diskless (network FS), 1 GbE MPI
- Measured 5.7–7.5 W per node depending under load
- 2 W for the ARM alone (0.5 W for the dualcore CPU), plus other board components, USB, GbE, blinking LEDs (0.5 W!!!)

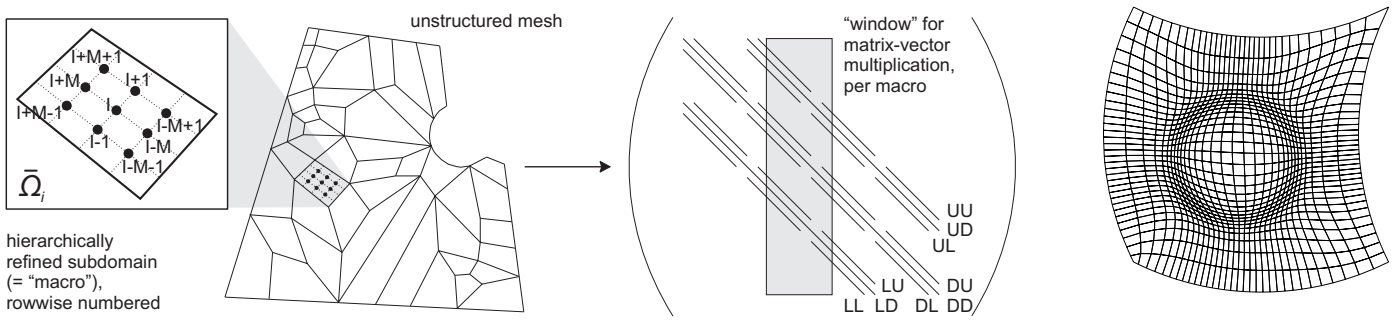
Porting effort

- Standard Linux OS with GNU compiler and debugger toolchain
- Main limitation: instruction issue rate $\Rightarrow$ sophisticated cache-blocking strategies counterproductive, bookkeeping and index overhead for deeper nested loops
- Serial codes only reach half of the memory bandwidth per node!

FEAST - parallel geometric multigrid

Globally unstructured locally structured grids

- Global macro-mesh: unstructured, flexible
- Local micro-meshes: structured (logical TP-structure), fast
- Numerical linear algebra: sequences of operations on structured data
- Cache-friendly and locality-maximising



Test case

- Poisson on cyclinderflow-mesh, Krylov-MG with strong smoother
- Essentially only limited by memory bandwidth

SPECFEM3D - seismic wave propagation modeling

Strong form of the wave equation as IVP

$$
\begin{aligned}
\rho \partial_t^2 u &= \nabla \cdot \sigma + f && \text{in } \Omega \times T \\
\sigma \cdot n &= \bar{t} && \text{on } \Gamma_N \\
u &= g && \text{on } \Gamma_D \\
u|_{t=0} &= u_0 && \forall x \in \Omega \\
\partial_t u|_{t=0} &= u_1 && \forall x \in \Omega
\end{aligned}
$$

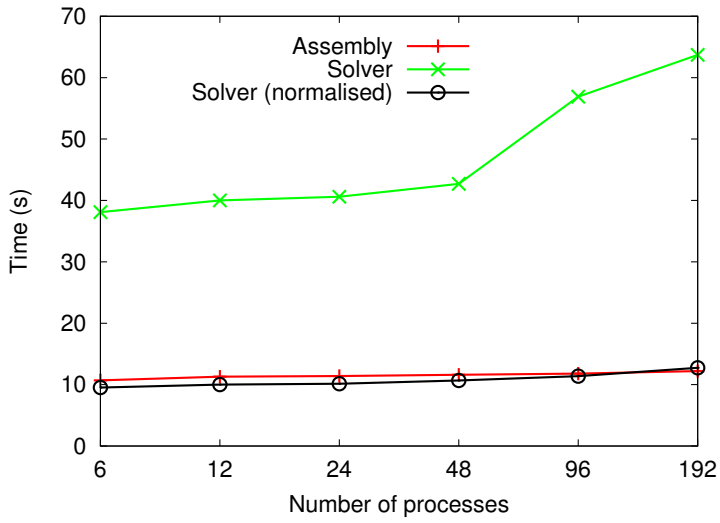
Discretisation

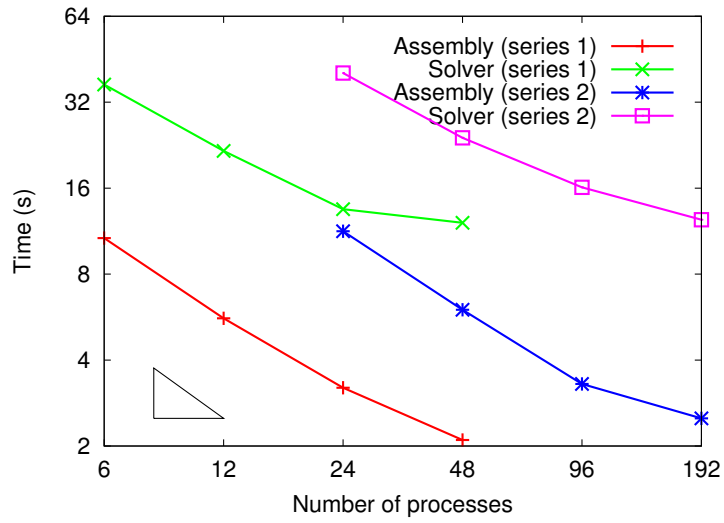
- High-order spectral element method, unstructured hexahedral mesh
- Leads to diagonal mass matrix, no linear solves necessary
- Second order explicit Newmark time-stepping

Limiting factors

- Mostly compute-bound

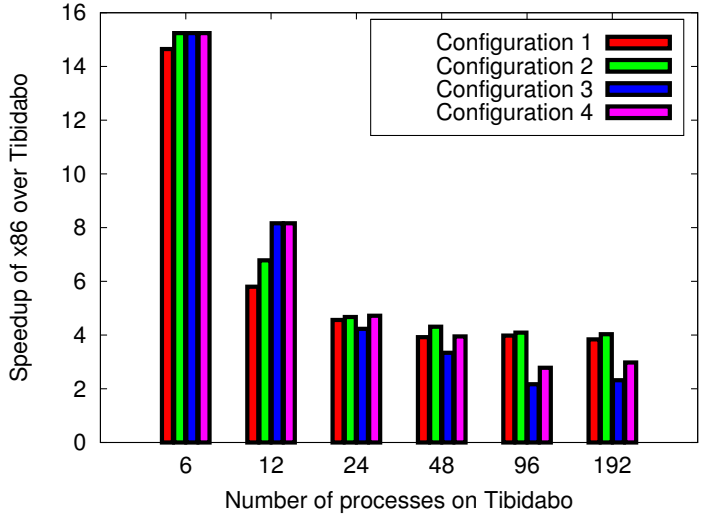
Scalability: FEAST

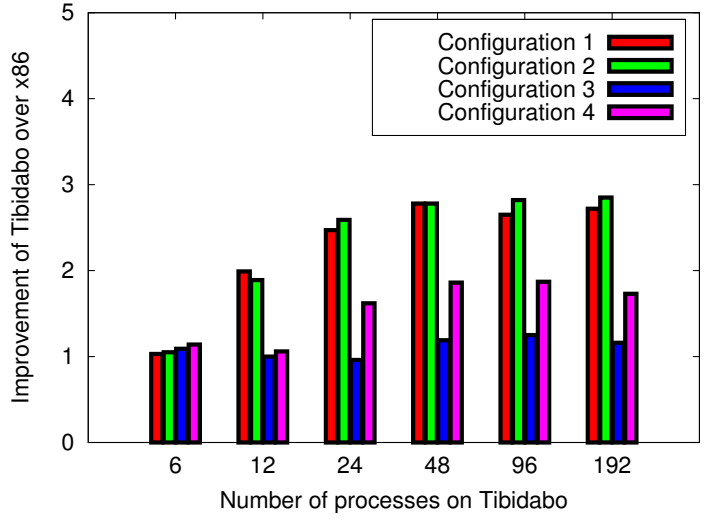




- Weak (left) and strong (right) scaling
- Weak scaling as expected (bump is a granularity effect from 4 to 5 solver iterations)
- Strong scaling quite ok (gigE and a multilevel method)

Power-performance analysis: FEAST

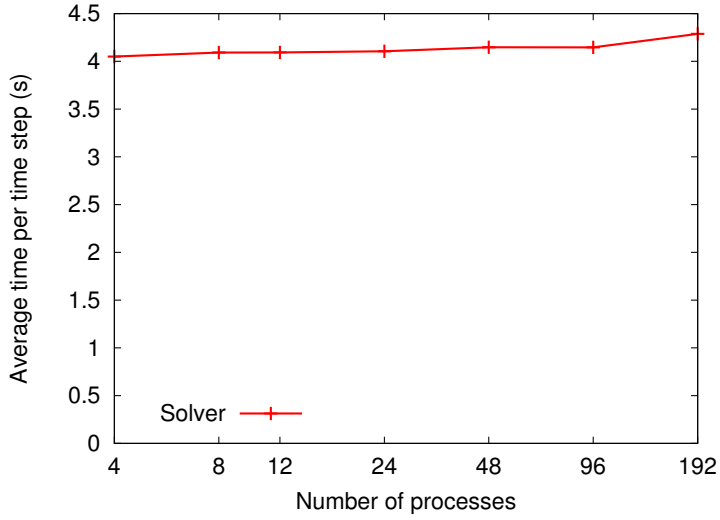


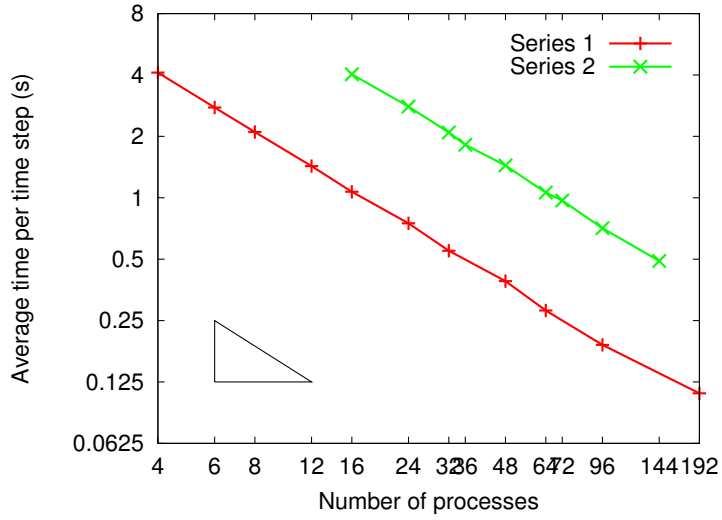


- Speedup x86 over ARM (left), improvement energy-to-solution ARM over x86 (right)
- Always more beneficial to use the ARM cluster
- As long as ≥ 2 x86 nodes are necessary, slowdown only 2–4

reference system: 32 2-way 4-core Nehalem system, 24 GB per node
(1) same load per core and partitioning (6 x86 cores/node), (2) re-partition for 8 cores/node, (3) as few x86 nodes as possible, (4) twice of that

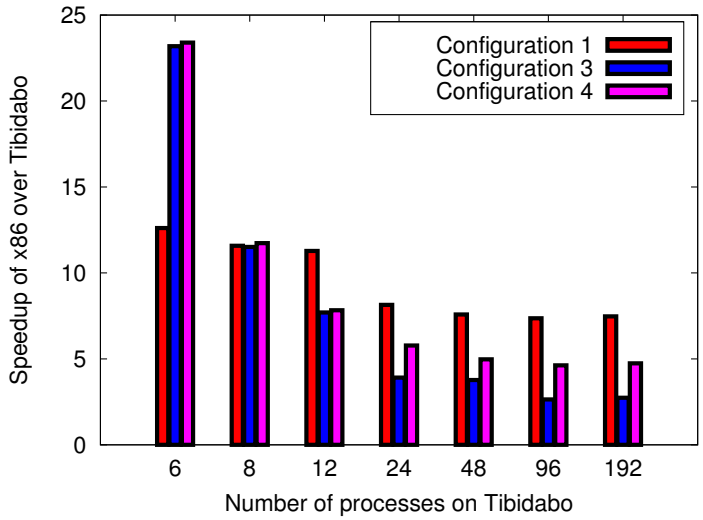
Scalability: SPECFEM3D_GLOBE

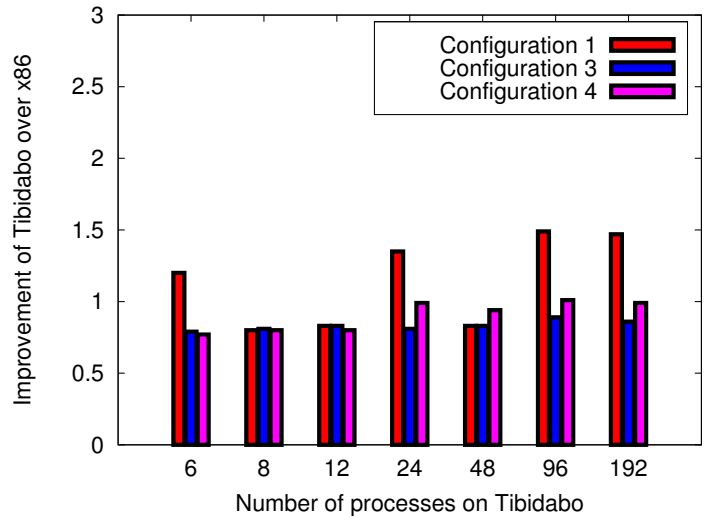




- Weak (left) and strong (right) scaling
- Both perfect (explicit in time, compute-bound, non-blocking MPI)

Power-performance analysis: SPECFEM3D_GLOBE





- Speedup x86 over ARM (left), improvement energy-to-solution ARM over x86 (right)
- Not always more beneficial to use the ARM cluster due to many small dense matrix matrix multiplications

Summary

Summary

- Power is the root cause of current and mid-term hardware evolution
- Energy efficiency will become the ultimate quality criterion
- Energy efficiency is quite hard to define exactly
- Weak and strong scaling essential even for moderate-size problems to compensate for slower execution
- Low-power architectures can be promising in terms of 'energy-to-solution' now already

Acknowledgments

Collaborative work with

- M. Geveler, D. Ribbrock (TU Dortmund)
- D. Komatitsch (CNRS Marseille)
- A. Ramirez, N. Rajovic, N. Puzovic (Barcelona Supercomputing Center)

Papers, links, tutorials and other stuff...

- `http://www.mathematik.tu-dortmund.de/~goeddeke`

Funding

- DFG Priority Program 1648: 'Software for Exascale Computing'

More acknowledgments

Acknowledgment:
today's Peta-op/s machines

10^{12} neurons @ 1 KHz = 1 PetaOp/s

**1.4 kilograms, 20 Watts **

[Shamelessly stolen from David Keyes]