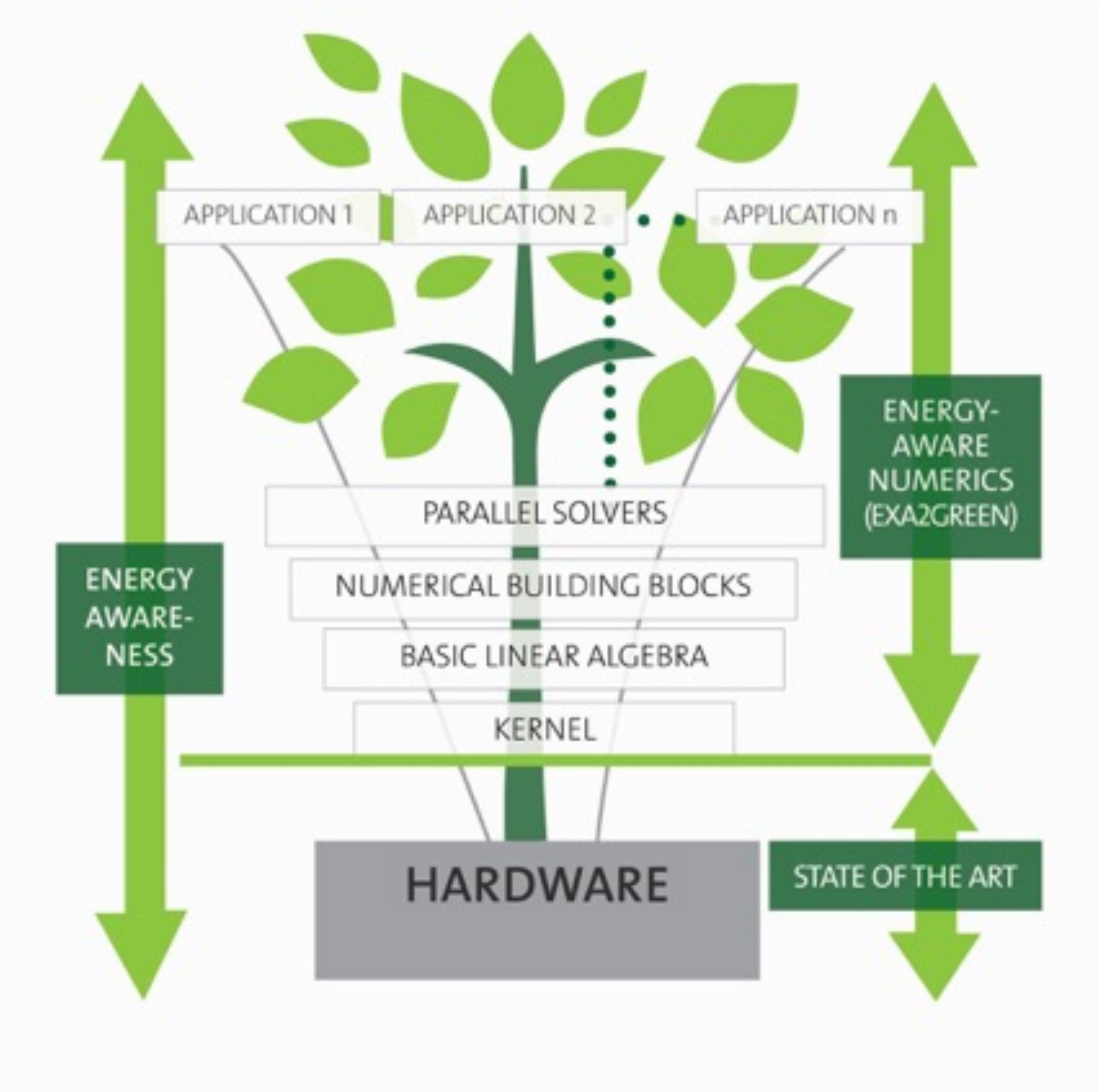
- Part 3 -
Energy Aware Numerics

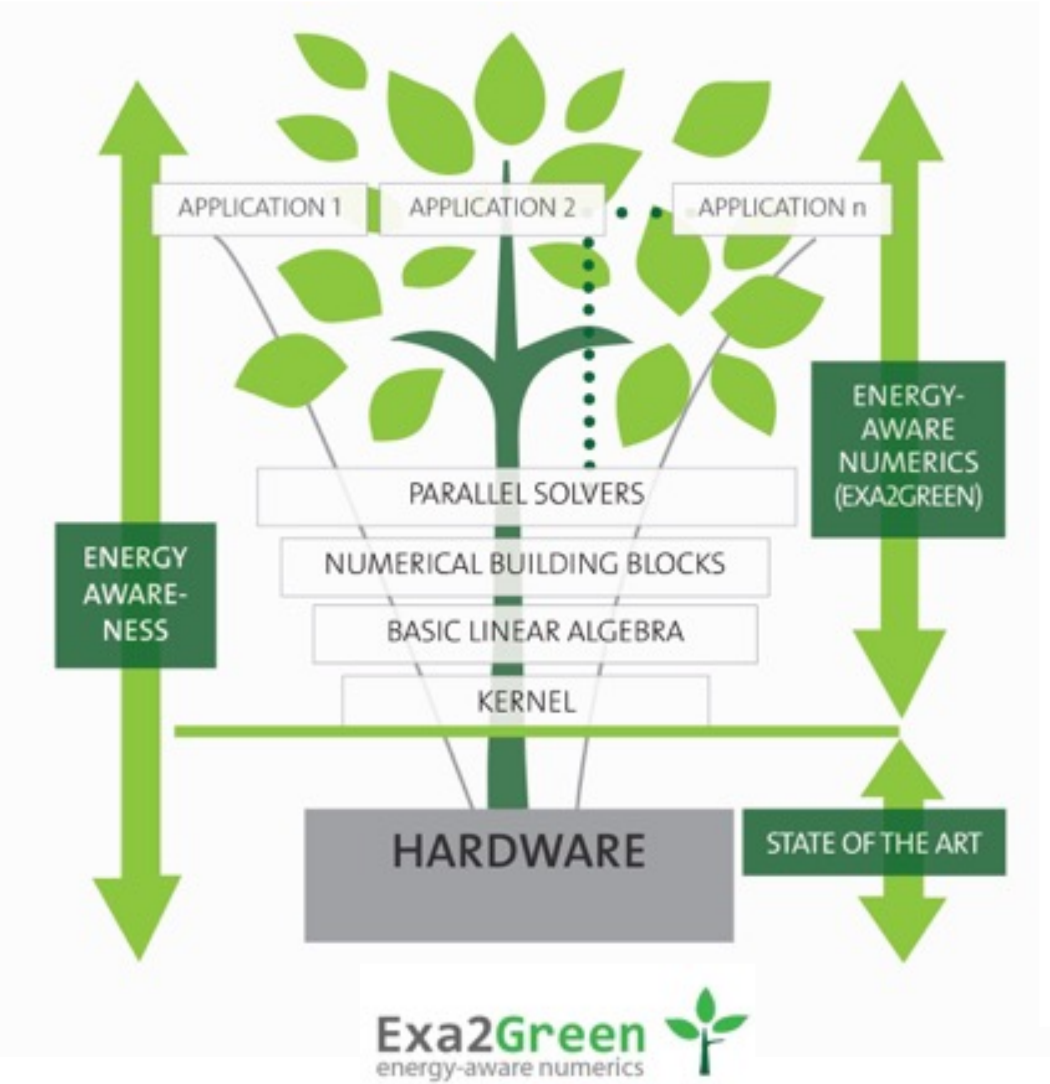
Vincent Heuveline

The Challenge



Exa2Green energy-aware numerics

Exa2green



Engineering Mathematics and Computing Lab	UHEI
Steinbeis Europa Zentrum	SEZ
Scientific Computing Group	UHAM
High Performance Computing and Architectures Group	UJI
Swiss National Supercomputing Centre	ETH Zurich
IBM Research Division	IBM
Institute for Meteorology and Climate Research	KIT

Objectives

Smart algorithms
Develop new smart algorithms using energy-efficient software models.

Profiling and new metrics
Develop an advanced and detailed power consumption monitoring and profiling for quantitative assessment and analysis of the energy profile of algorithms.

Power-aware scheduling
Smart and power-aware scheduling and hardware adaption technology for HPC.

Proof of concept

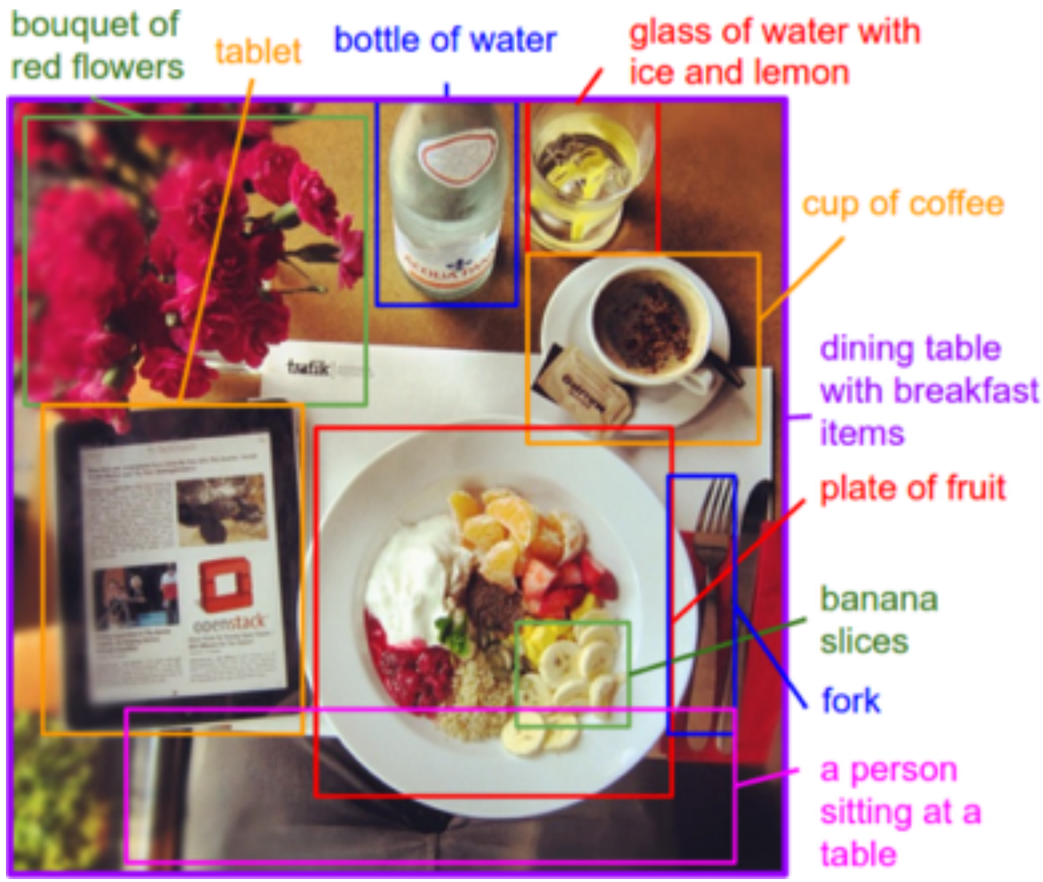
Fast, accurate and energy-efficient computation of the exponential function

Applications:

- Neural networks
- Fourier transform
- Statistics, probability
- Radioactive decay
- Population models

Existing techniques:

- Power series
- IEEE-754 manipulation
- Look-up tables



Karpathy, Fei-Fei: Deep Visual-Semantic Alignments for Generating Image Descriptions. CVPR 2015

Fast, accurate and energy-efficient computation of the exponential function

Existing techniques: Truncated power series

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots$$

Method:
Compute partial sum

$$e^x \approx \sum_{n=0}^{N} \frac{x^n}{n!}$$

PRO: uses arithmetic, can exploit SIMD
PRO: flexible accuracy
CON: very slow convergence, too many FLOP for high accuracy

Fast, accurate and energy-efficient computation of the exponential function

Existing techniques: Truncated power series

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \ldots$$

Method:
Compute partial sum

$$e^x \approx \sum_{n=0}^{N} \frac{x^n}{n!}$$

PRO: uses arithmetic, can exploit SIMD
PRO: flexible accuracy
CON: very slow convergence, too many FLOP for high accuracy

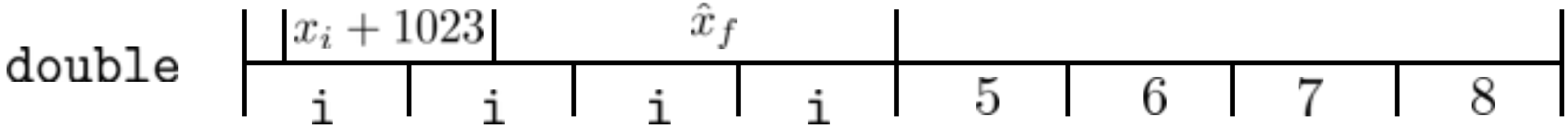
Fast, accurate and energy-efficient computation of the exponential function

Existing techniques: IEEE-754 manipulation

Schraudolph 1999

$$e^x = 2^{x/\ln 2} = 2^{x_i + x_f} = 2^{x_i} \cdot 2^{x_f}$$

$$\texttt{int i} = 2^{20}\left(\frac{x}{\ln 2} + 1023\right) + C \;\texttt{;}$$



$$\Longrightarrow (-1)^s(1+m)2^{c-1023} = (1+\hat{x}_f)2^{x_i} \approx e^x$$

where $\hat{x}_f =$ 20 most significant digits of $x_f + 2^{-20}C$

Fast, accurate and energy-efficient computation of the exponential function

New technique:
Malossi, Ineichen, Bekas, Curioni: Fast Exponential Computation on SIMD Architectures. WAPCO 2015

Patent application no. CH920140048US1

$$e^x = 2^{x \log_2(e)} = 2^{x_i + x_f} = 2^{x_i} \cdot 2^{x_f} = \left(1 + x_f - C(x_f)\right) 2^{x_i}$$

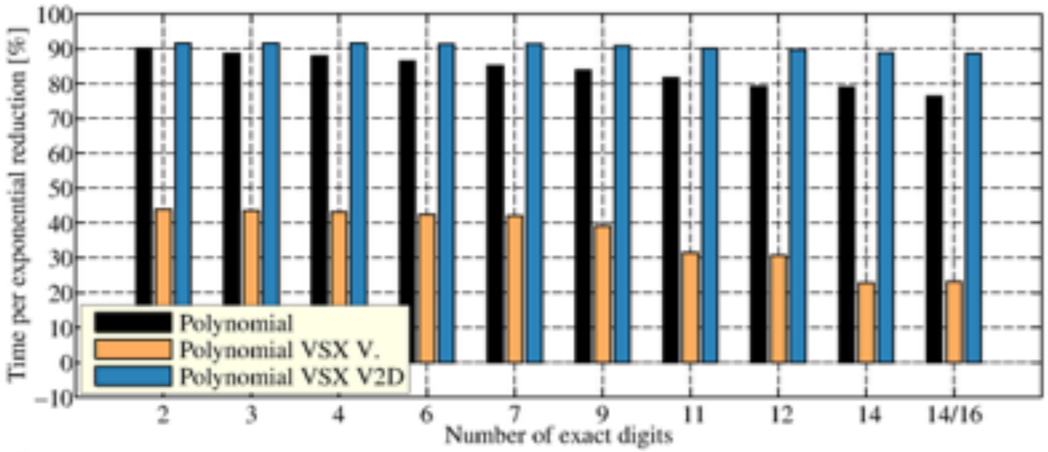
exact correction $\quad C(x_f) = 1 + x_f - 2^{x_f}$

polynomial approximation $\quad C_n(x_f) = a_0 + a_1 x_f + a_2 x_f^2 + \ldots a_n x_f^n$

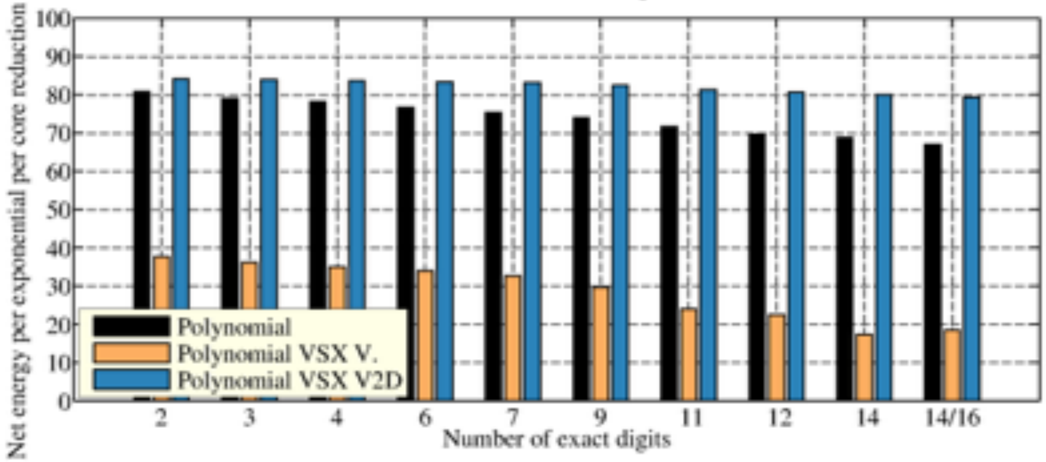
return $\quad \exp_n(x) := \left(1 + x_f - C_n(x_f)\right) 2^{x_i} \approx e^x$

Fast, accurate and energy-efficient computation of the exponential function

time percent reduction



energy percent reduction

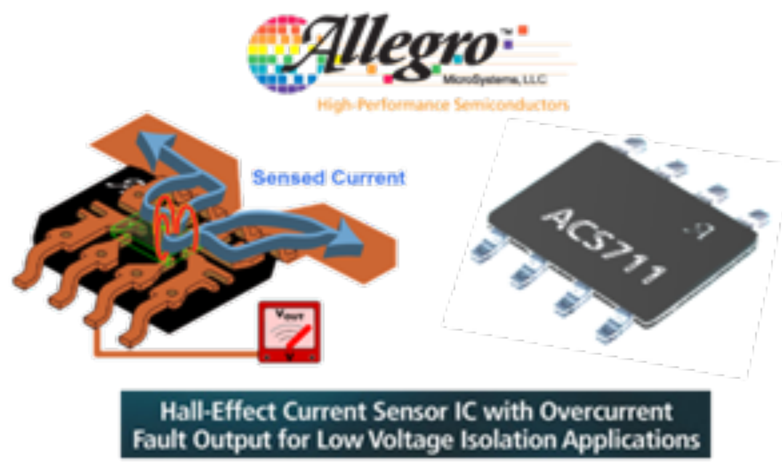


Malossi, Ineichen, Bekas, Curioni: Fast Exponential Computation on SIMD Architectures. WAPCO 2015

ArduPower: A new low-cost internal wattmeter

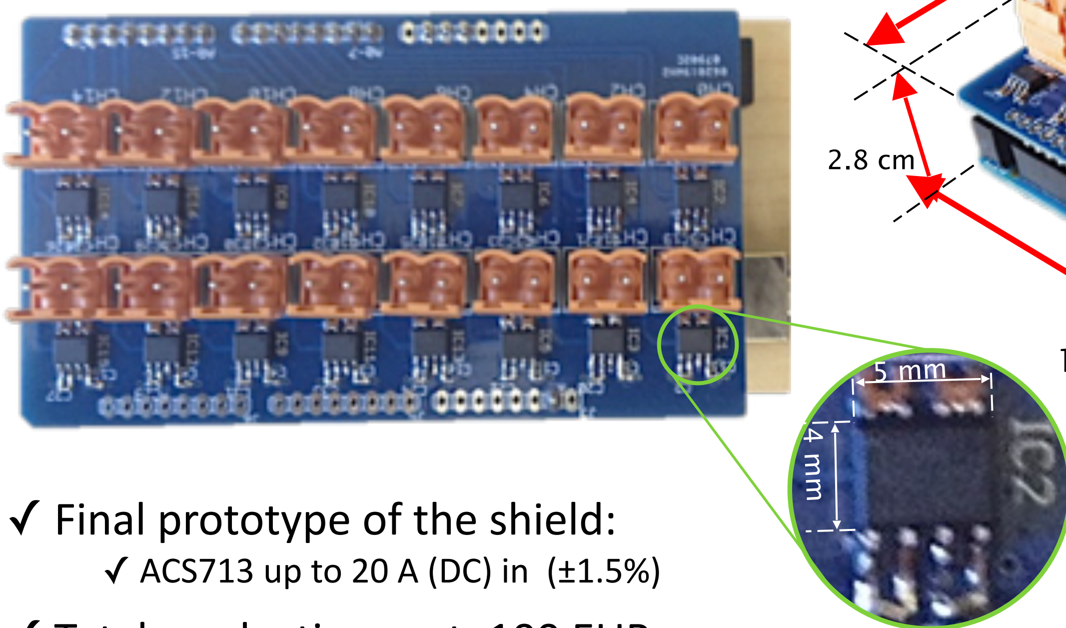
- ✓ **Objective:** Measure internally the power consumption of computers

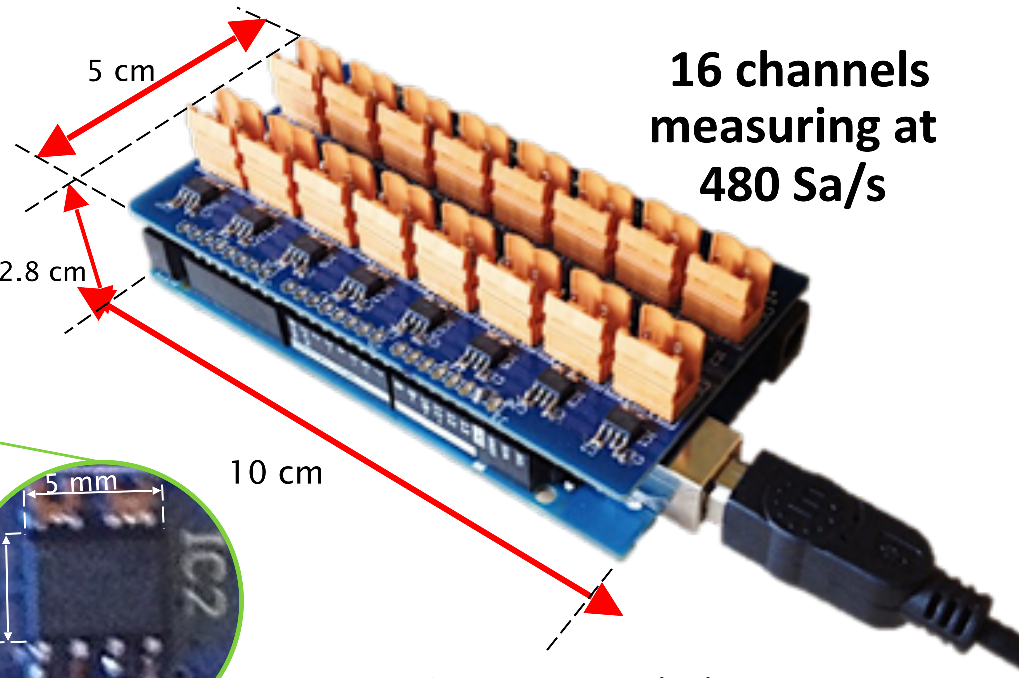
  Problem: internal power meters are expensive and difficult to use (e.g., National Instruments DAS)
- ✓ **Requirements**: Build-up a **accurate, small** and **cheap** new wattmeter device
  - ✓ *Microcontroller*: Arduino Mega 2560 with 16 analogue channels ~50 EUR
  - ✓ *Sensors*: Allegro Hall-Effect IC sensor ACS series (accuracy ±5%) ~2 EUR/chip
- ✓ **Solution**: A new shield for Arduino Mega with Allegro Hall-Effect sensors!



ArduPower: A new low-cost internal wattmeter

The prototype:





16 channels measuring at 480 Sa/s

- ✓ Final prototype of the shield:
  - ✓ ACS713 up to 20 A (DC) in (±1.5%)
- ✓ Total production cost: 100 EUR
- ✓ PCB circuits available on demand
- ✓ Integration into PMLib!

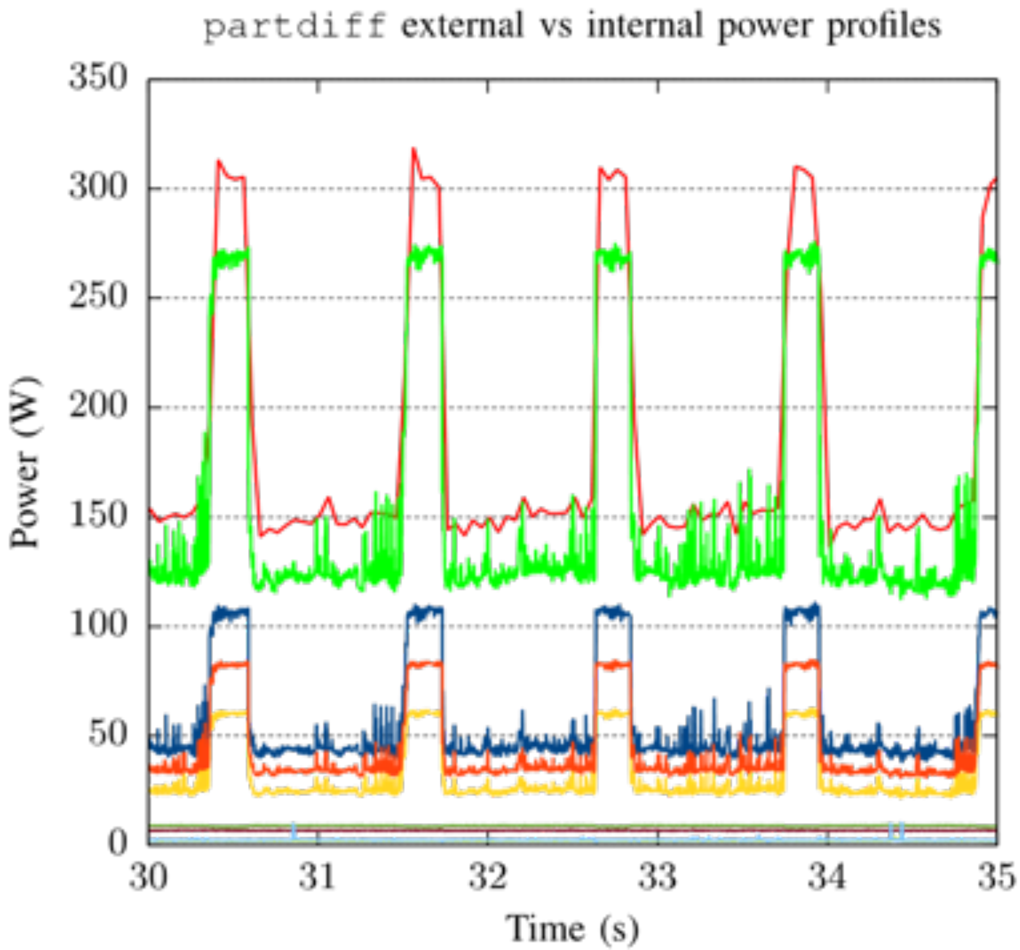
Design from G. Fabregat (UJI)

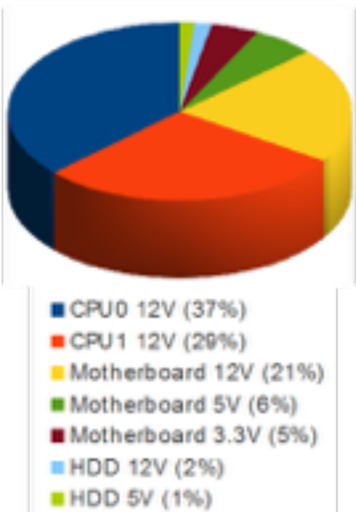
USB Serial port (but it could also work via Ethernet)

Dolz, Heidari, Kuhn, Ludwig, Fabregat: ArduPower: A Low-cost Wattmeter to improve Energy Efficiency of HPC Applications. 6th Int. Conf. Green & Sustainable Computing, 2015

ArduPower: A new low-cost internal wattmeter

✓ An example using a PDE solver: (Partial differential equation solver for Gauss-Seidel and Jacobi Method)

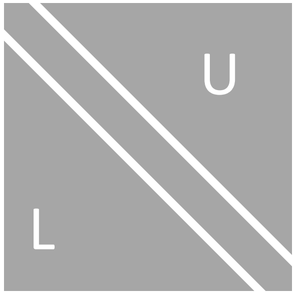




Courtesy of M.R. Heidari, Hamburg University

Splitting Method

matrix splitting $A = L + D + U =$



linear equation system

$$(L + D + U)x = b$$

$$Dx = b - (L + U)x$$

$$x = D^{-1}b - D^{-1}(L + U)x$$

Jacobi Iteration

$$x = D^{-1}b - \underbrace{D^{-1}(L+U)}_{B}x$$

iteration matrix B

Jacobi iteration

$$x_i^{k+1} = \frac{1}{a_{ii}}\Big(b_i - \sum_{j \neq i} a_{ij} x_j^k\Big)$$

- parallel component updates within one iteration
- synchronization between iterations
- converges if $\rho(B) < 1$

Asynchronous Iteration

Jacobi method $$x_i^{k+1} = \frac{1}{a_{ii}}\Big(b_i - \sum_{j\neq i} a_{ij} x_j^k\Big)$$

update function

shift function

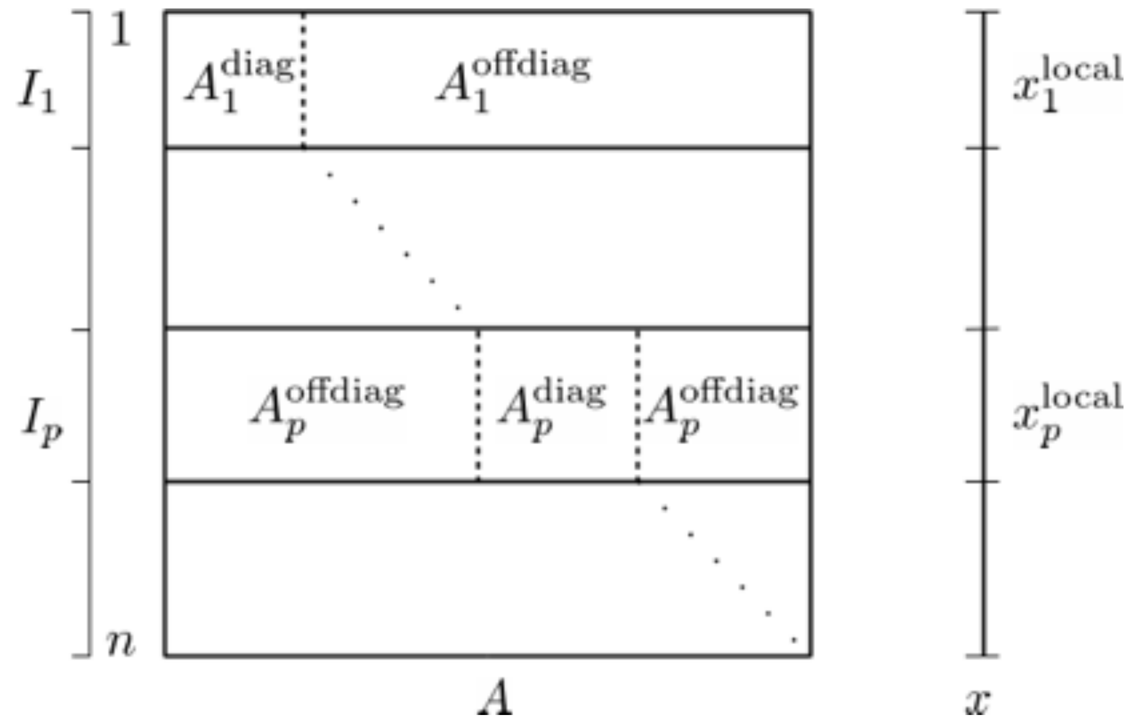
Asynchronous iteration

$$x_i^{k+1} = \begin{cases} \frac{1}{a_{ii}}\Big(b_i - \sum_{j\neq i} a_{ij} x_j^{k-s(k,j)}\Big) & \text{if } i = u(k) \\ x_i^k & \text{if } i \neq u(k) \end{cases}$$

- introduce asynchronism, reduce communication
- parallel component updates, less synchronization
- converges if $\rho(|B|) < 1$

Block-asynchronous Iteration

divide matrix
into blocks

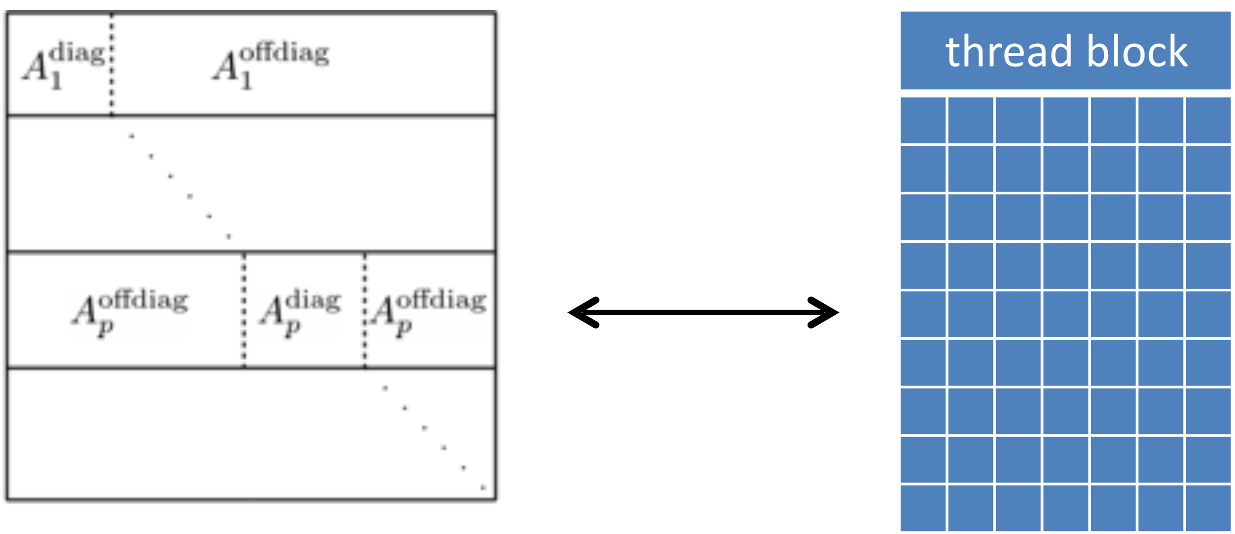


block update $x_p^{\mathrm{local}} \leftarrow D_p^{-1}\left[b_p^{\mathrm{local}} - A_p^{\mathrm{diag}} x_p^{\mathrm{local}} - A_p^{\mathrm{offdiag}} x_p^{\mathrm{non\text{-}local}}\right]$

Block-asynchronous Iteration on GPU

	SMX = streaming multiprocessor	cores per SMX	max. thread blocks per SMX	max. threads per thread block
Tesla K40	15	192	16	1024

- matrix blocks correspond to thread blocks
- synchronous Jacobi iteration within the thread block
- asynchronous iteration with other thread blocks

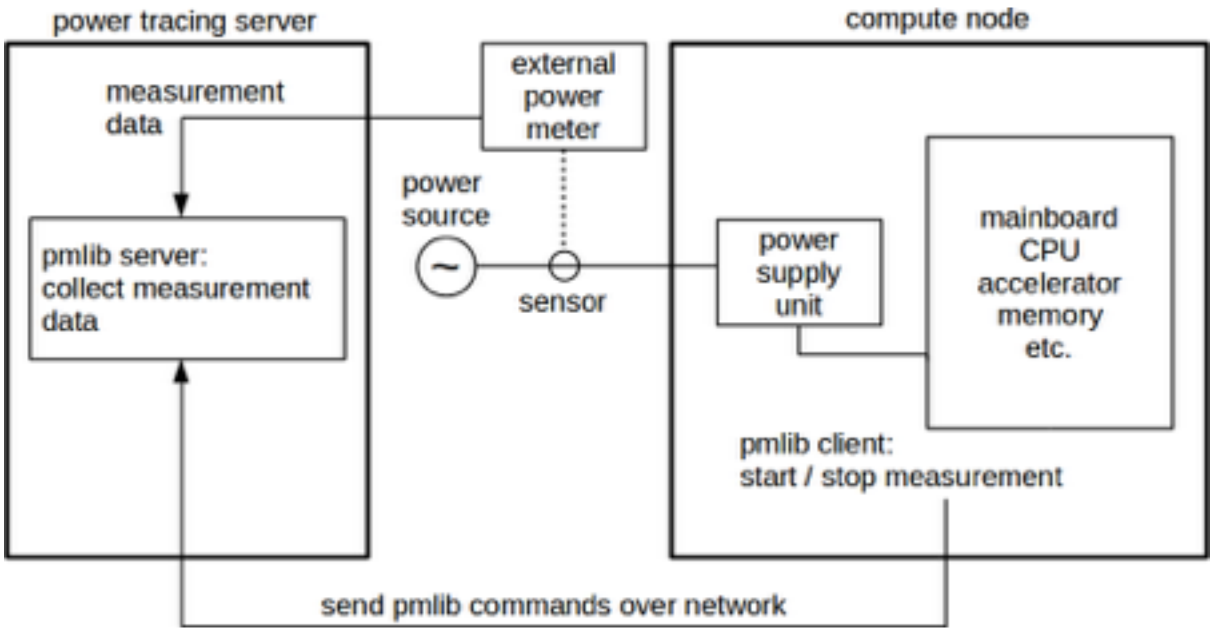


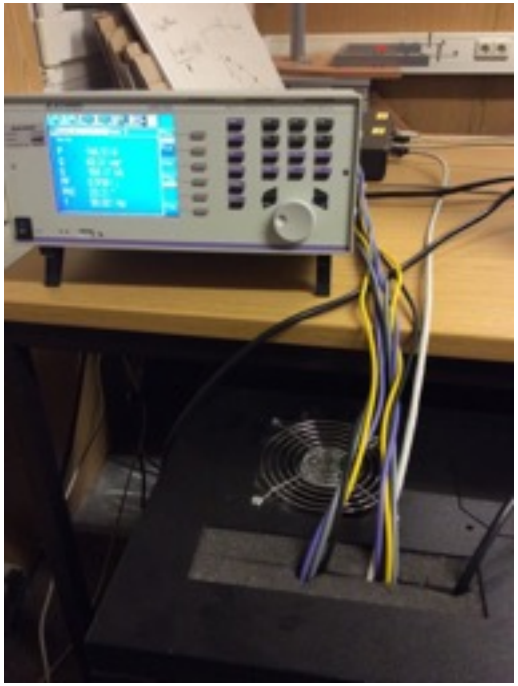
Related Works

- Rosenfeld 1969:
  experiments on „chaotic relaxation“ for use on „parallel-processor computing systems“, simulation of current distribution in electric networks
- Chazan & Miranker 1969:
  first rigorous analysis of „chaotic relaxation“, convergence theory, examples of divergence
- Overviews of „asynchronous iteration“:
  e.g. Bertsekas & Tsitsiklis 1989, Frommer & Szyld 2005
- Anzt et al. 2011, 2013:
  block-asynchronous iteration on GPU-accelerated systems

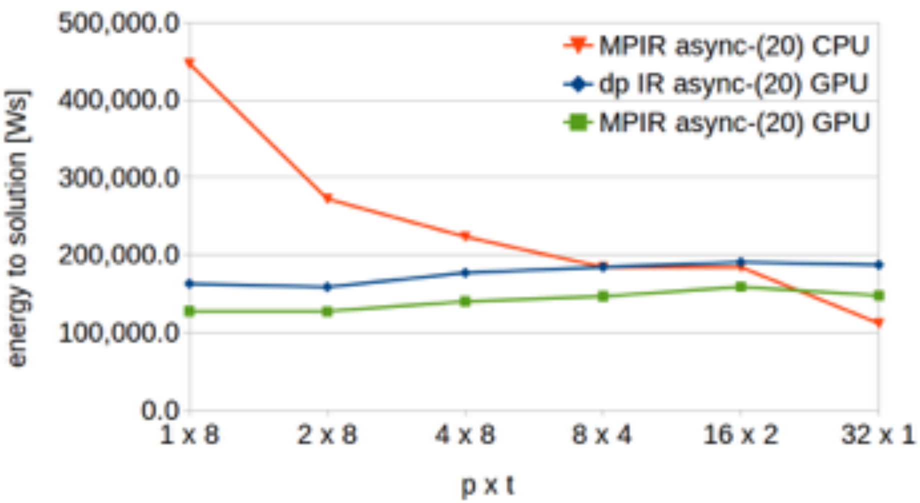
Experimental setup

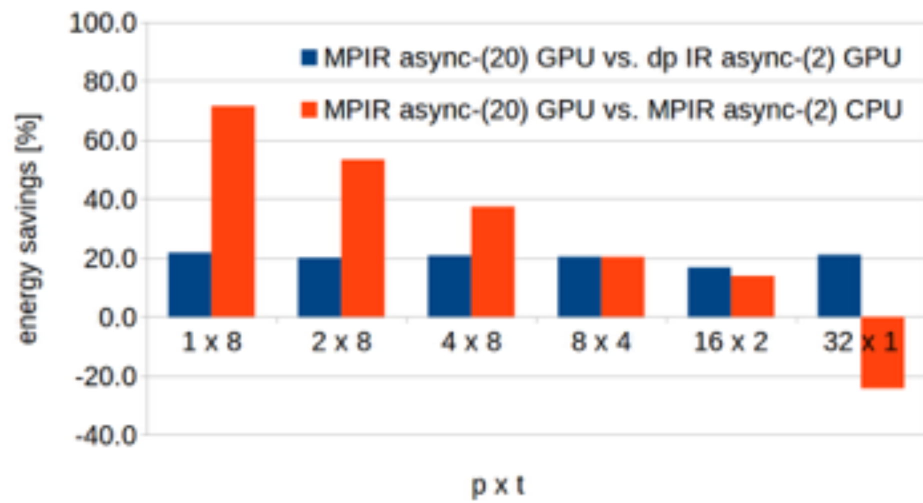
- energy measurement
  Zimmer Electronics Systems LMG450
  pmlib – power measurement library





Energy-to-solution & savings





- large energy savings of more than 50 % for GPU usage on small host systems, i.e. cases 1 x 8 and 2 x 8
- moderate saving of 20 % - 40 % for GPU usage in cases 4 x 8 to 16 x 2
- strong host systems can outperform GPU w.r.t. runtime and energy, see case 32 x 1

Thank you