ALEA:Abstraction-Level Energy Accounting and Optimization for Many-core Programming Languages

Lev Mukhanov

Queen's University of Belfast

01.01.16 ECIT

Queen's University Belfast

Outline

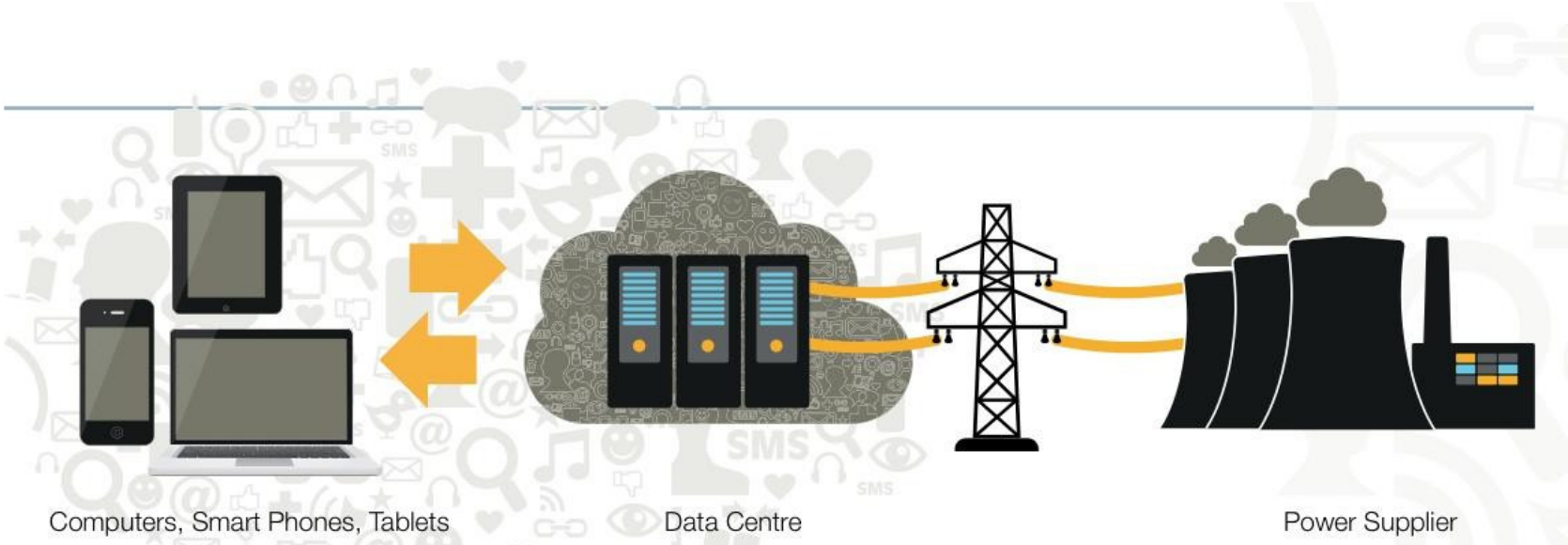
- **Introduction**
- Probabilistic approach
- Challenges
- ALEA implementation
- Use-cases
- DIVIDEND and UNISERVER

Consortium

The project is funded by:

Consortium:

Data centers are the new polluters

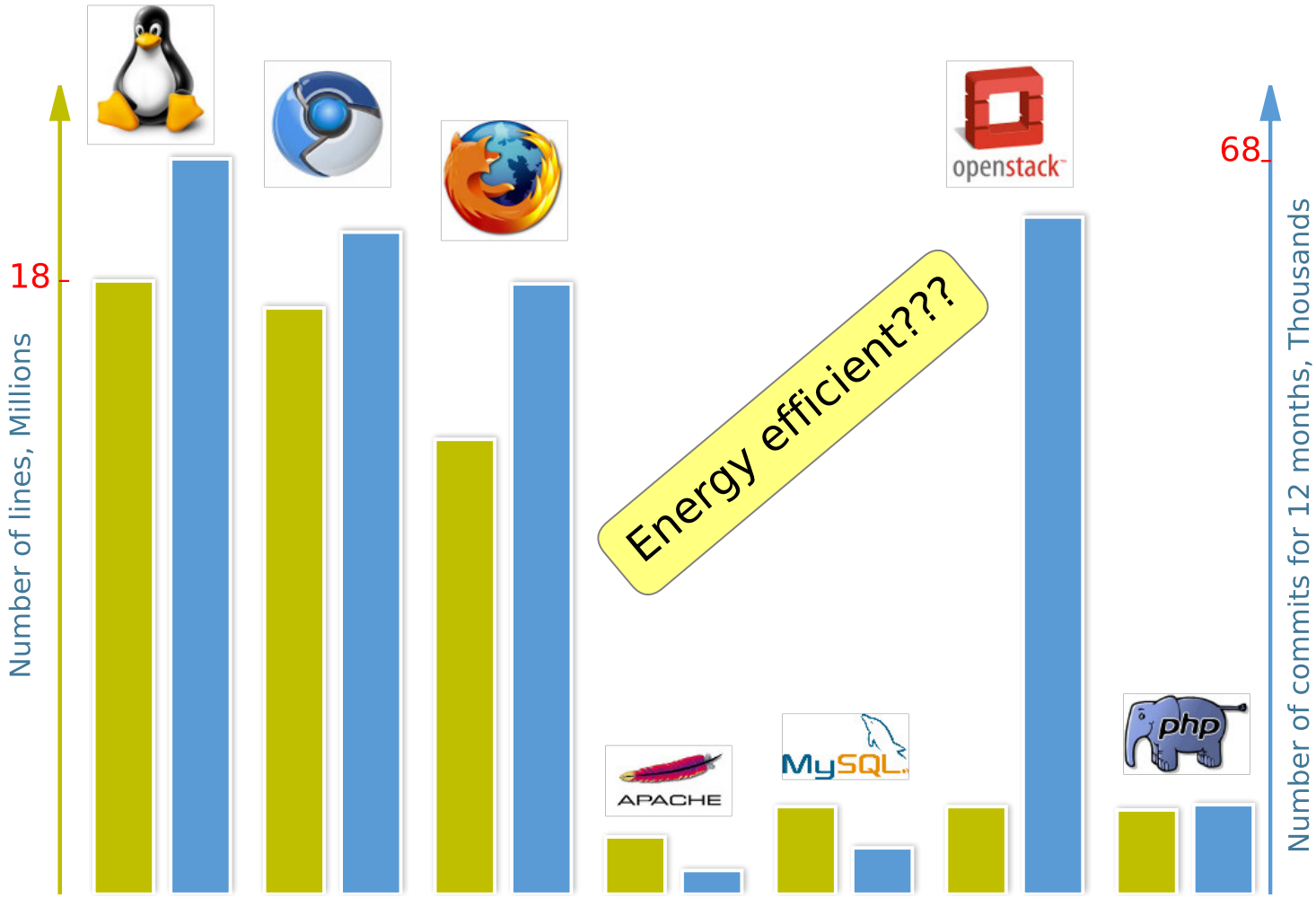


- the world's ICT(Information Communications Technologies) ecosystem now approaches 10% of world electricity generation(2013)

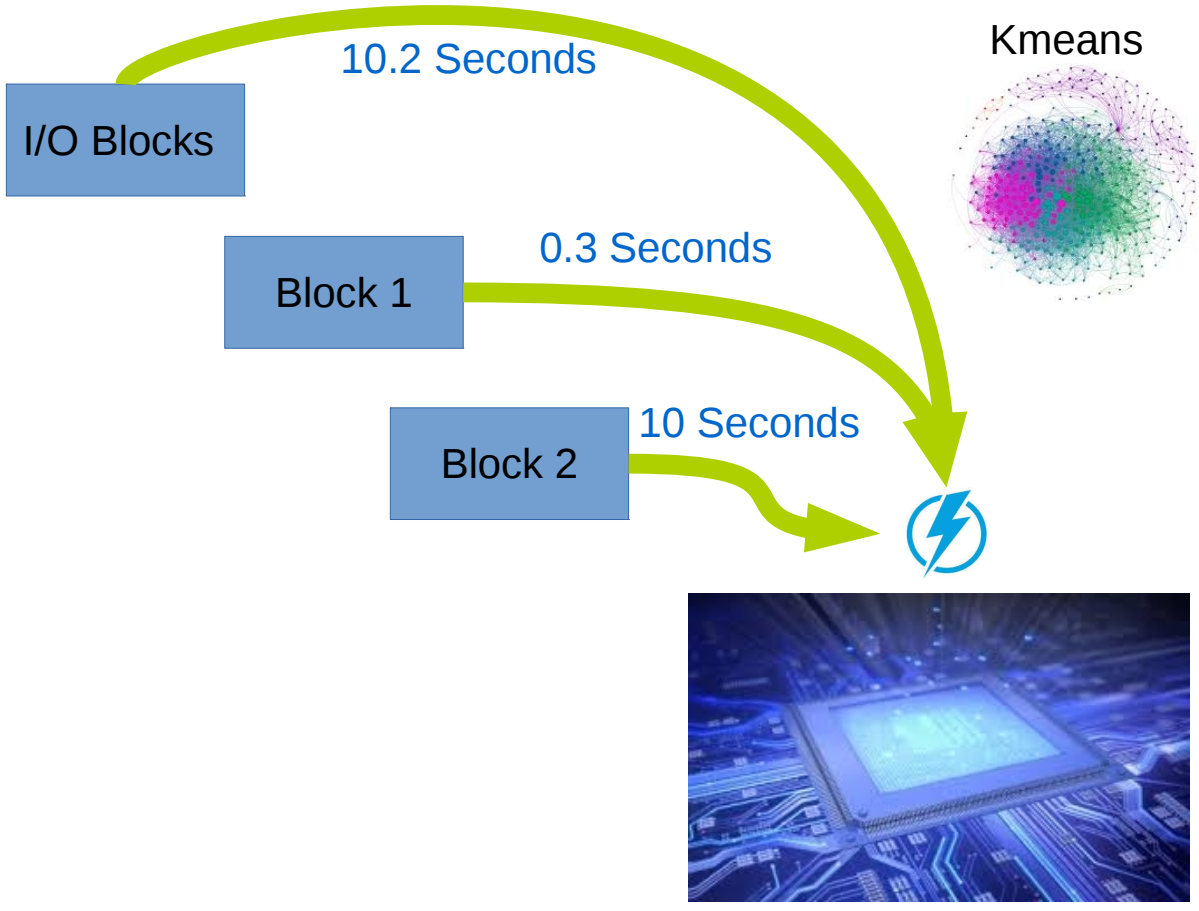
Data centers are the new polluters

Year	End-use energy(B kWh)	Elec. Bills (US,$B)	Power plants (500 MW)	CO2(US) (Million MT)
2013	91	$9.0	34	97
2020	139	$13.7	51	147
2013-2020 increase	47	$4.7	17	50

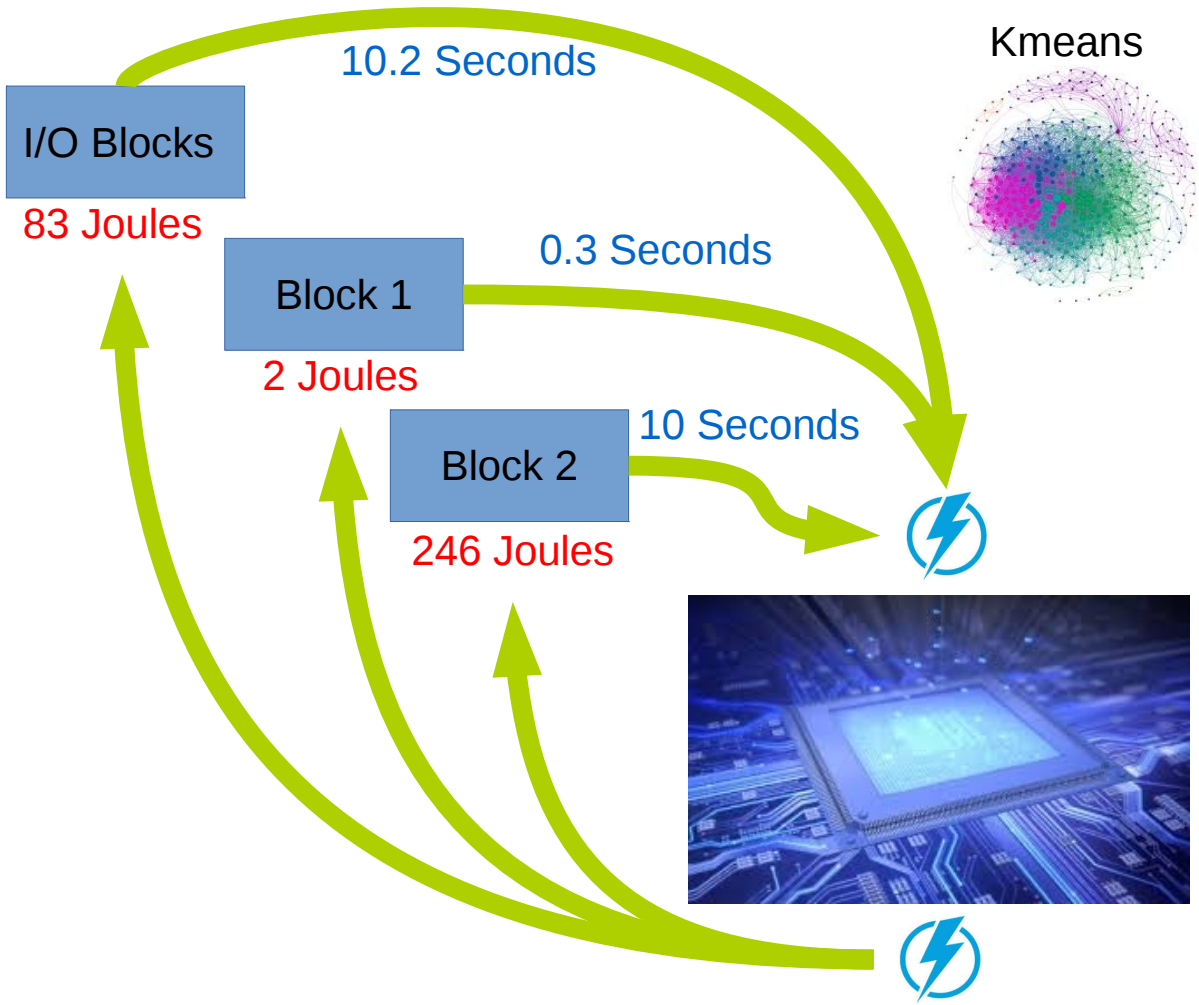
Energy optimization challenge



How workloads affect energy?

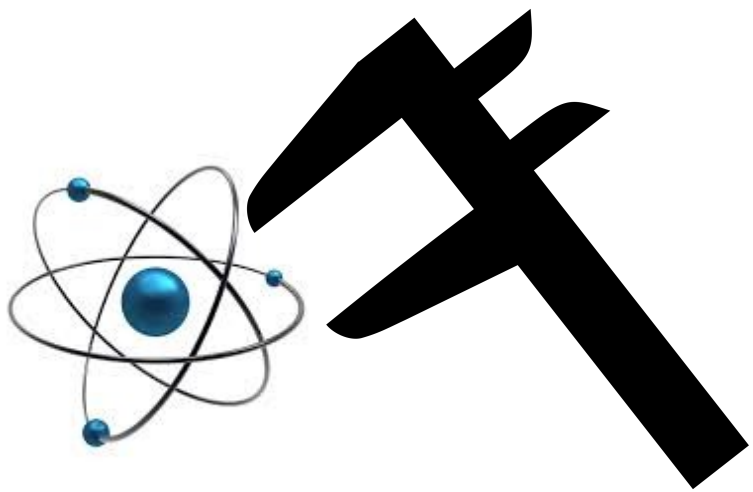


How workloads affect energy?



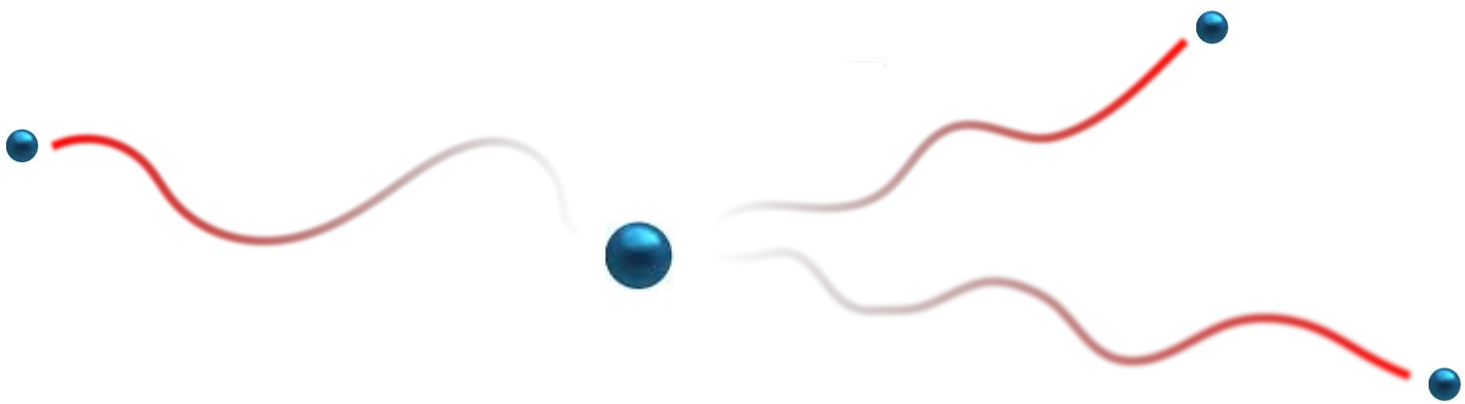
Fine-grain energy profiling challenges

- Coarse-grained power/energy meters
- Any measurements bias real energy
- Overhead introduced by measurements is critical



Fine-grain energy profiling challenges

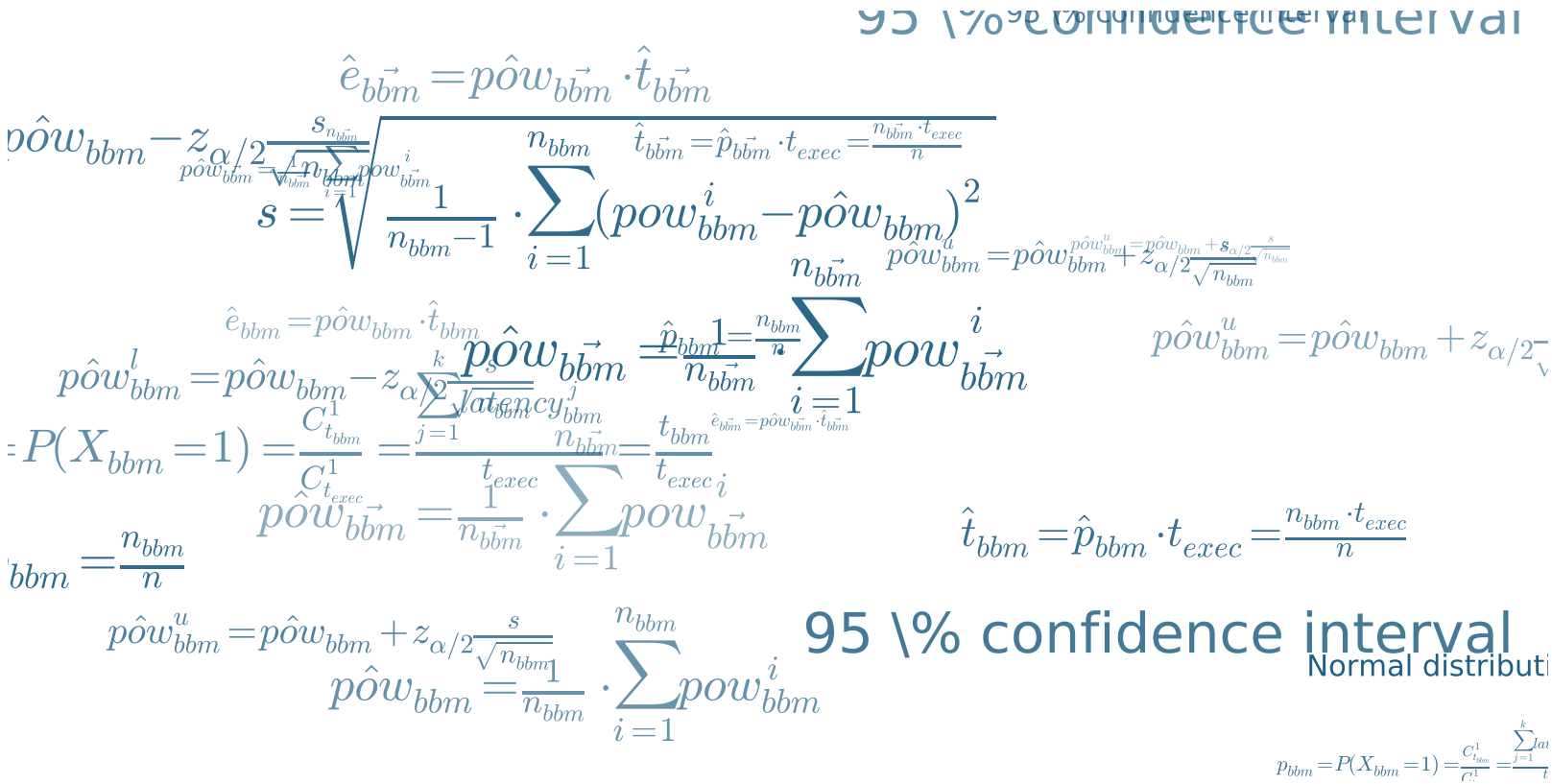
- Coarse-grained power/energy meters
- Any measurements bias real energy
- Overhead introduced by measurements is critical



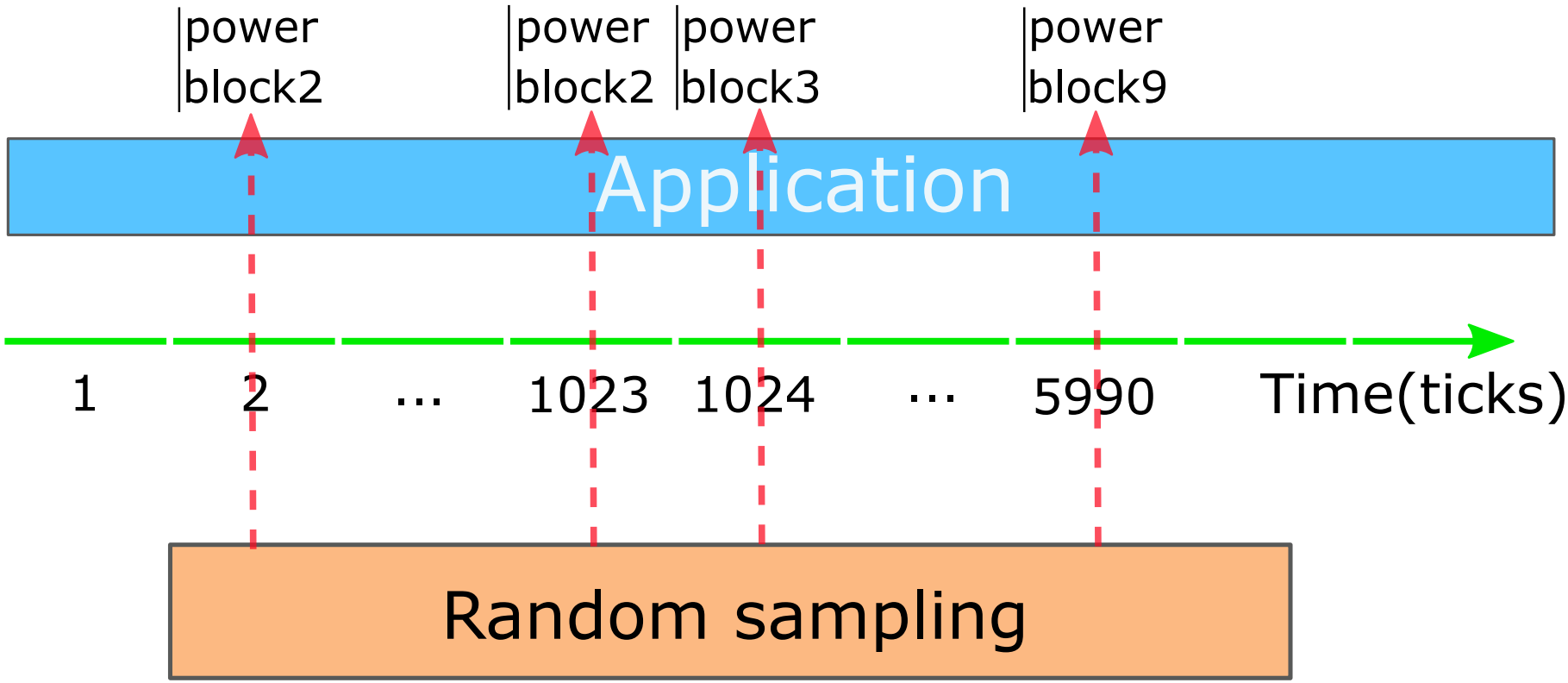
Outline

- Introduction
- **Probabilistic approach**
- Challenges
- ALEA implementation
- Use-cases
- DIVIDEND and UNISERVER

Probabilistic model



Random sampling



Probabilistic model

- Execution time of a block:

$$time_{block} = p_{block} \cdot time_{application} \tag{1}$$

- Estimation of $\hat{p}_{block}$ using sampling
- Estimation of execution time:

$$\hat{time}_{block} = \hat{p}_{block} \cdot time_{application} \tag{2}$$

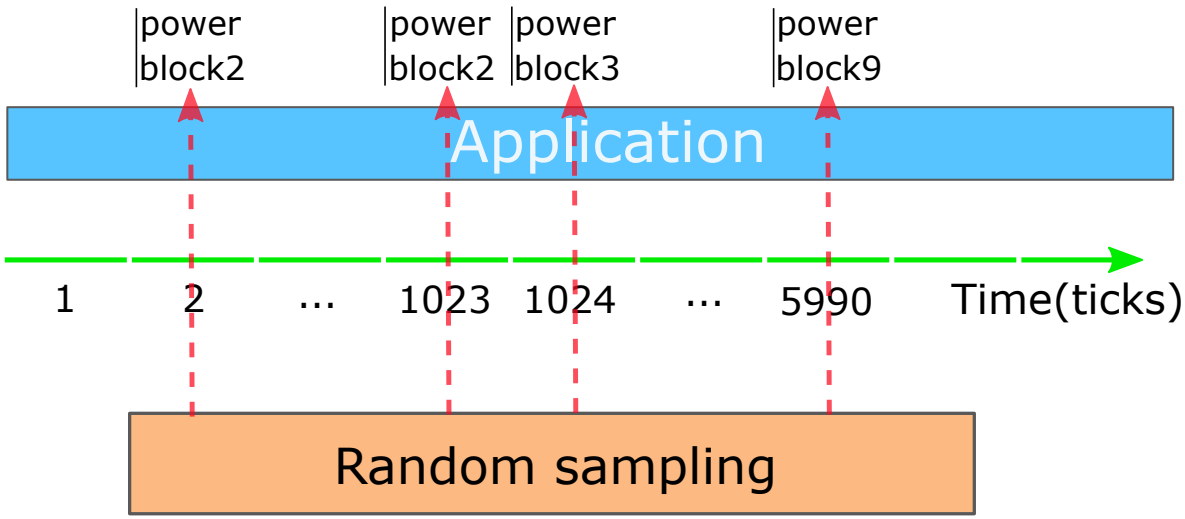
- Power measurements and a sample are taken simultaneously to estimate $\hat{power}_{block}$
- Estimation of energy consumption:

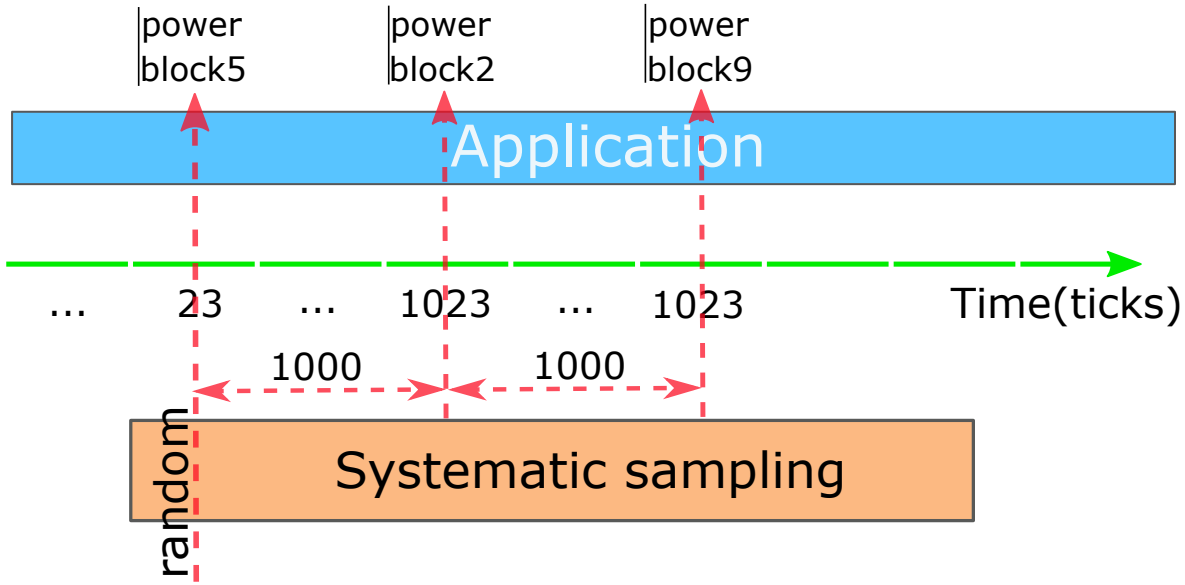
$$\hat{energy}_{block} = \hat{power}_{block} \cdot \hat{time}_{block} \tag{3}$$

Outline

- Introduction
- Probabilistic approach
- **Challenges**
- ALEA implementation
- Use-cases
- DIVIDEND and UNISERVER

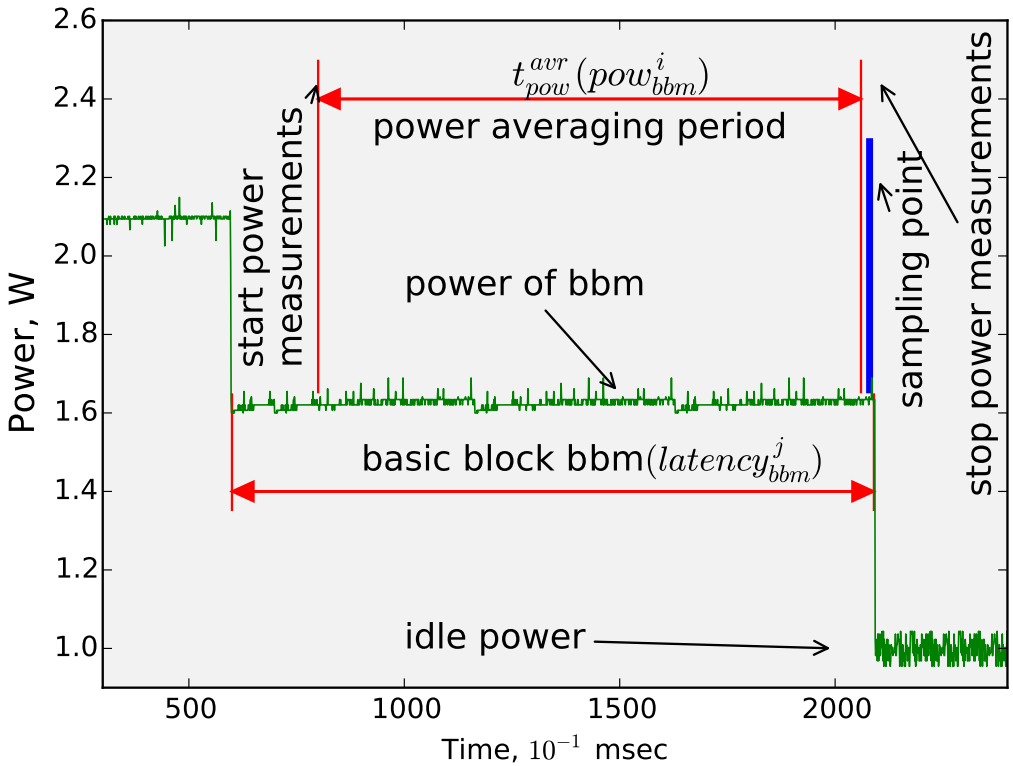
Random sampling challenge



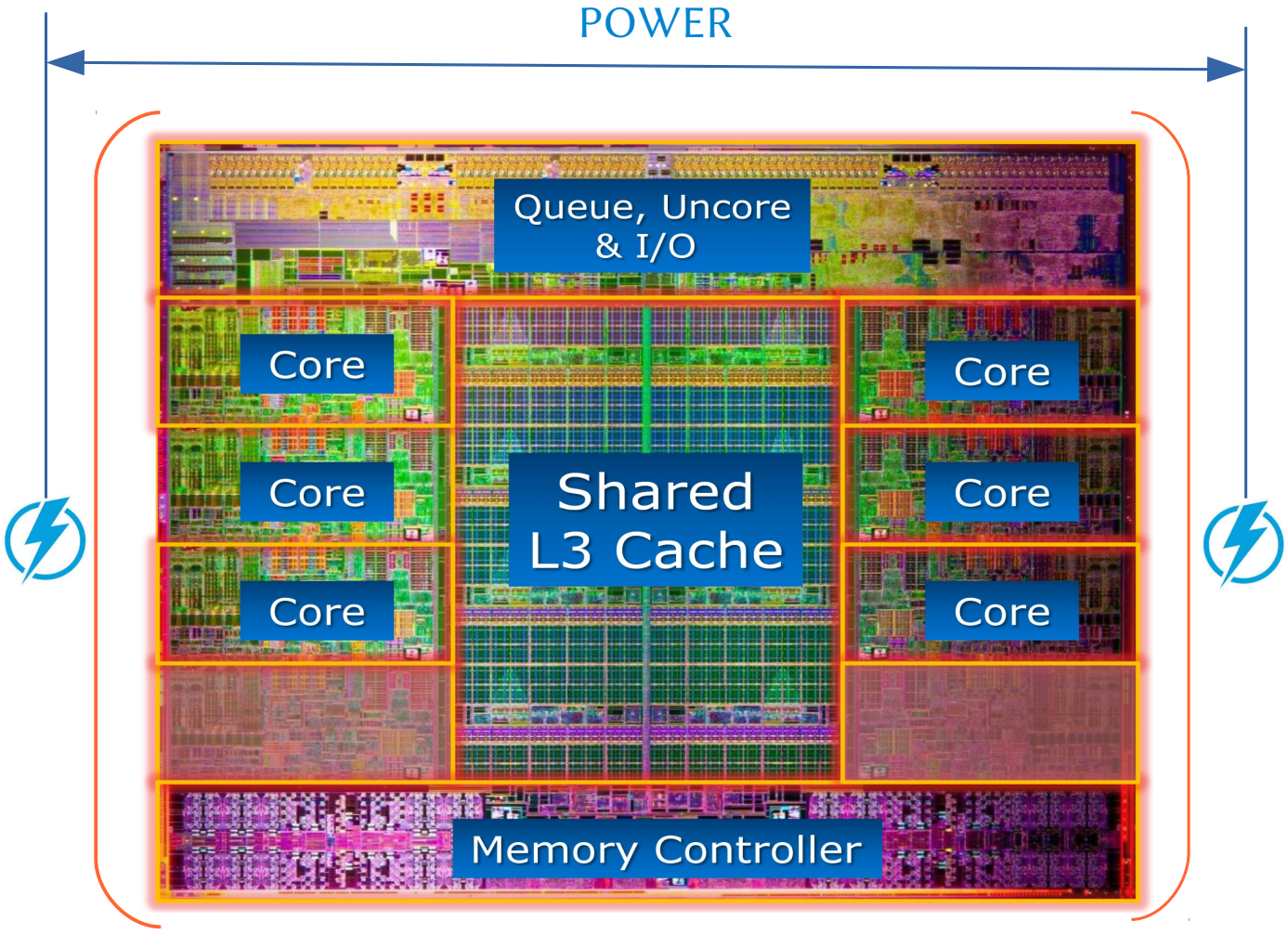


Power measurements challenge

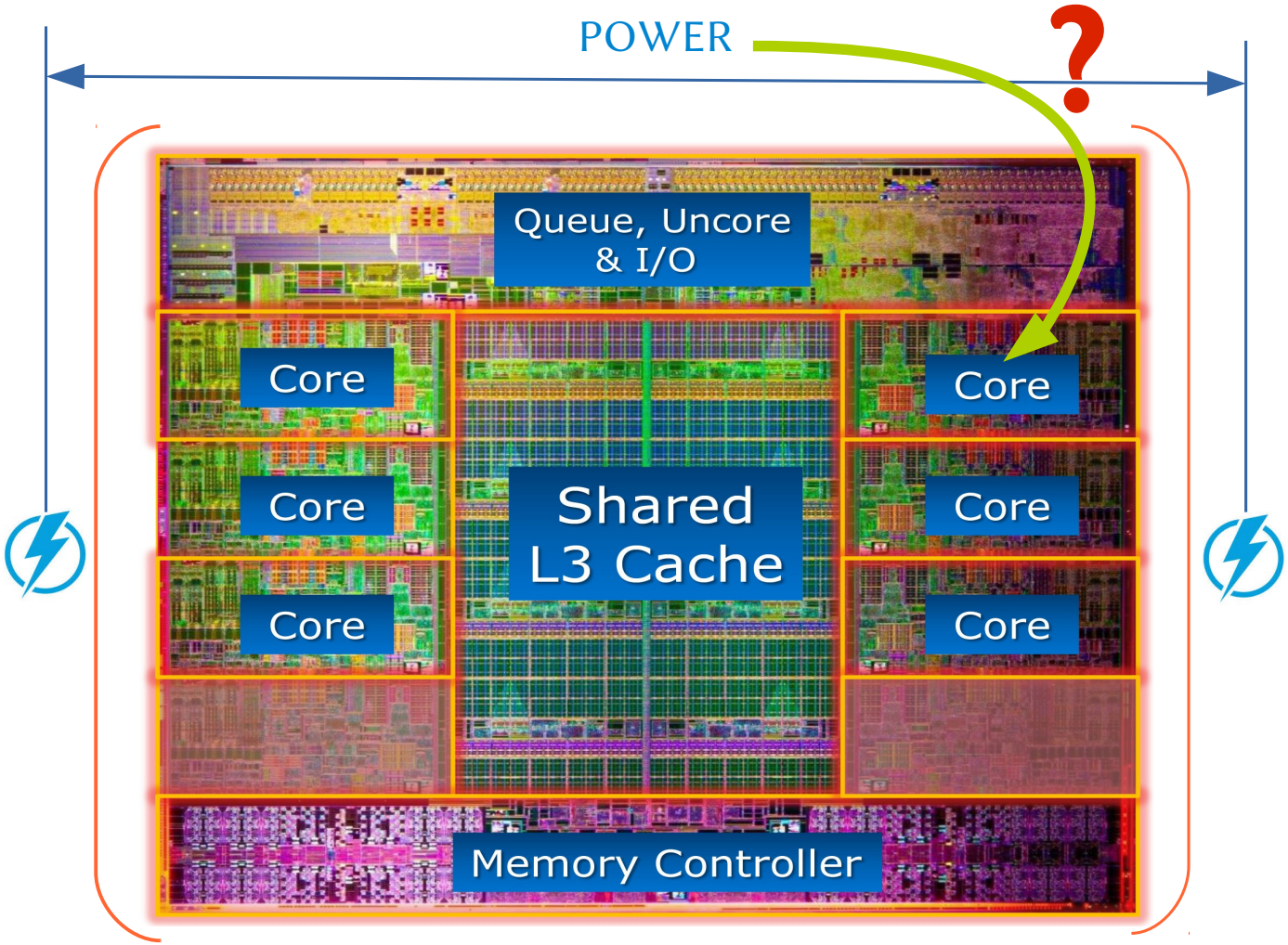
- The proposed model implies instant power measurements
- **BUT the power measurements are averaged over some period!!!**



Parallel profiling challenge



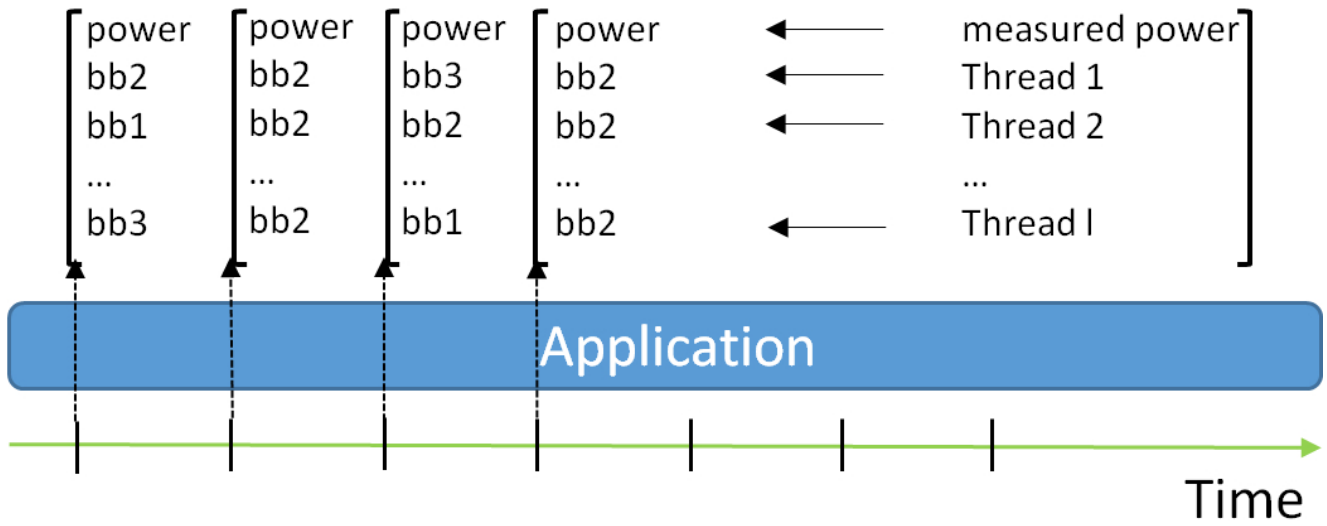
Parallel profiling challenge



Profiling of parallel applications

- How to apportion power/energy between threads?
-  Basic block vector(BBV) $\vec{bbm}$:

$$\vec{bbm} = bb_{thread_1}, bb_{thread_2}, ..., bb_{thread_l} \tag{4}$$



Results of profiling

Code blocks	Samples	Time	Energy	Power
core1:cb100,core2:cb100	163	21.99 +/-0.44	67.95 +2.71/-2.65	3.09 +/-0.06
core1:cb100,core2:sys	94	0.95 +/-0.19	1.66 +0.42/-0.39	1.75 +/-0.08
core1:cb161,core2:cb161	91	0.92 +/-0.18	2.29 +0.54/-0.51	2.48 +/-0.07

How to present the results of profiling?

Power 31.623466 , Thread 45919, core 0: RIP=0x4ebcb8 Thread 45931, core 2: RIP=0x52ee52 Thread 45932, core 4: RIP=0x52a3b8 Thread 45933, core 6: RIP=0x520796 Thread 45934, core 8: RIP=0x5230cc Thread 45935, core 10: RIP=0x51037d Thread 45936, core 1: RIP=0x51d91c Thread 45937, core 11: RIP=0x504861 Thread 45938, core 7: RIP=0x5a0870 Thread 45939, core 5: RIP=0x527fee Thread 45940, core 9: RIP=0x50d648 Thread 45941, core 3: RIP=0x4f0f20

Power 32.003697 , Thread 45919, core 0: RIP=0x4f0f42 Thread 45931, core 2: RIP=0x4ebcaa Thread 45932, core 4: RIP=0x50a82d Thread 45933, core 6: RIP=0x581ef7 Thread 45934, core 8: RIP=0x5230aa Thread 45935, core 10: RIP=0x4ebcaa Thread 45936, core 1: RIP=0x50d658 Thread 45937, core 11: RIP=0x50d657 Thread 45938, core 7: RIP=0x51ae5c Thread 45939, core 5: RIP=0x5a0f05 Thread 45940, core 9: RIP=0x534268 Thread 45941, core 3: RIP=0x522ff4

Power 32.410092 , Thread 45919, core 0: RIP=0x4ebcbb Thread 45931, core 2: RIP=0x529de5 Thread 45932, core 4: RIP=0x52e29e Thread 45933, core 6: RIP=0x525c36 Thread 45934, core 8: RIP=0x52bff9 Thread 45935, core 10: RIP=0x51adc5 Thread 45936, core 1: RIP=0x51ad95 Thread 45937, core 11: RIP=0x4ebcaa Thread 45938, core 7: RIP=0x5230ef Thread 45939, core 5: RIP=0x52ef3e Thread 45940, core 9: RIP=0x51ad95 Thread 45941, core 3: RIP=0x4ebcaa

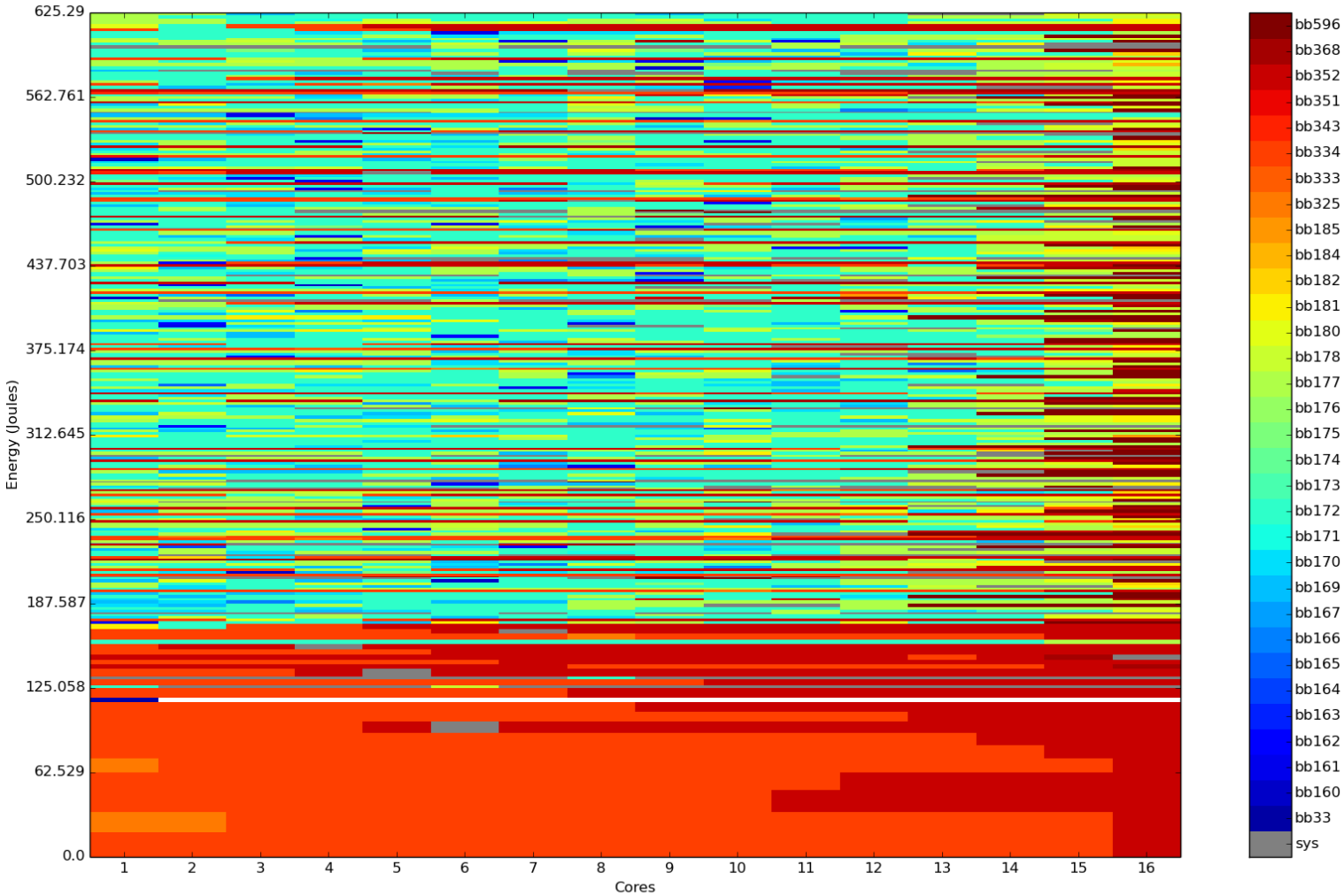
Power 31.610973 , Thread 45919, core 0: RIP=0x4ebcb8 Thread 45931, core 2: RIP=0x4eca0d Thread 45932, core 4: RIP=0x83413295 Thread 45933, core 6: RIP=0x4ebcaa Thread 45934, core 8: RIP=0x4ebcaa Thread 45935, core 10: RIP=0x4f0fbd Thread 45936, core 1: RIP=0x4ebcaa Thread 45937, core 11: RIP=0x520737 Thread 45938, core 7: RIP=0x50d635 Thread 45939, core 5: RIP=0x4fe390 Thread 45940, core 9: RIP=0x51304a Thread 45941, core 3: RIP=0x4ebcaa

Power 32.113810 , Thread 45919, core 0: RIP=0x50a8aa Thread 45931, core 2: RIP=0x52303c Thread 45932, core 4: RIP=0x5230a0 Thread 45933, core 6: RIP=0x4edd63 Thread 45934, core 8: RIP=0x5045d5 Thread 45935, core 10: RIP=0x4f0f0e Thread 45936, core 1: RIP=0x51d91c Thread 45937, core 11: RIP=0x4ec848 Thread 45938, core 7: RIP=0x516edb Thread 45939, core 5: RIP=0x4ebcaa Thread 45940, core 9: RIP=0x4fa502 Thread 45941, core 3: RIP=0x4ebcaa

Power 32.305270 , Thread 45919, core 0: RIP=0x4f4f96 Thread 45931, core 2: RIP=0x5230bb Thread 45932, core 4: RIP=0x4f1306 Thread 45933, core 6: RIP=0x4ebcbb Thread 45934, core 8: RIP=0x504861 Thread 45935, core 10: RIP=0x527ffe Thread 45936, core 1: RIP=0x51ae47 Thread 45937, core 11: RIP=0x5259bf Thread 45938, core 7: RIP=0x52303f Thread 45939, core 5: RIP=0x4ebcaa Thread 45940, core 9: RIP=0x523052 Thread 45941, core 3: RIP=0x4ebcaa

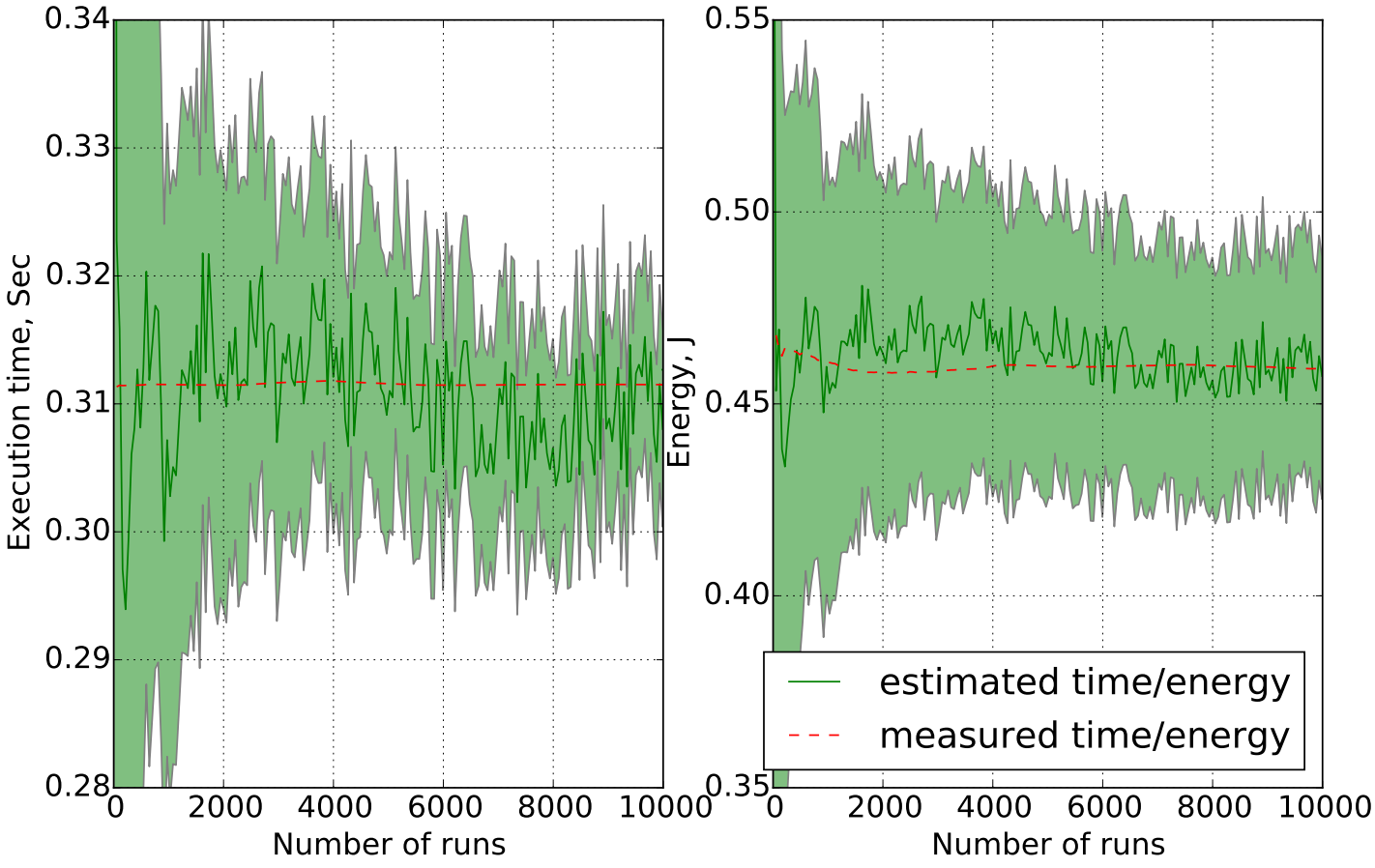
Power 64.399139 , Thread 45919, core 0: RIP=0x4ebcaa Thread 45931, core 2: RIP=0x52c10d Thread 45932, core 4: RIP=0x50d61e Thread 45933, core 6: RIP=0x513036 Thread 45934, core 8: RIP=0x513041 Thread 45935, core 10: RIP=0x51ad76 Thread 45936, core 1: RIP=0x50452f Thread 45937, core 11: RIP=0x4f0fb8 Thread 45938, core 7: RIP=0x50d614 Thread 45939, core 5: RIP=0x527fce Thread 45940, core 9: RIP=0x50d626

Visual Analytics



Accuracy challenge

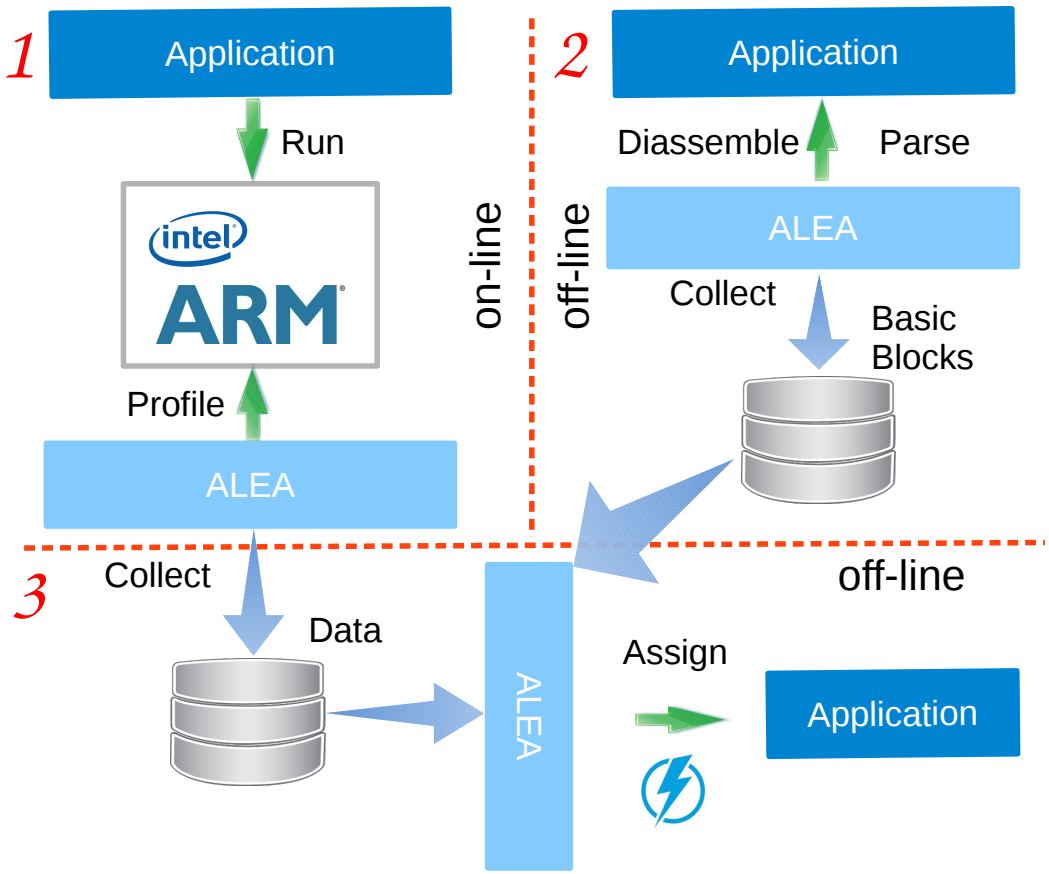
- How to increase accuracy $\Longrightarrow$ the number of samples?
-  **Multiple-run profiling**



Outline

- Introduction
- Probabilistic approach
- Challenges
- **ALEA implementation**
- Use-cases
- DIVIDEND and UNISERVER

Implementation



- DWARF is used to assign energy estimates to source code
- Architecture independent implementation(portable)
- Low overhead(1%) - suitable for on-line profiling

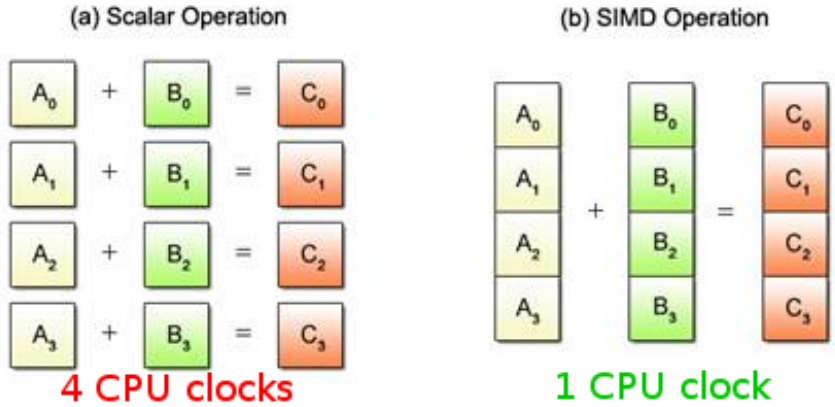
Outline

- Introduction
- Probabilistic approach
- Challenges
- ALEA implementation
- **Use-cases**
- DIVIDEND and UNISERVER

Energy

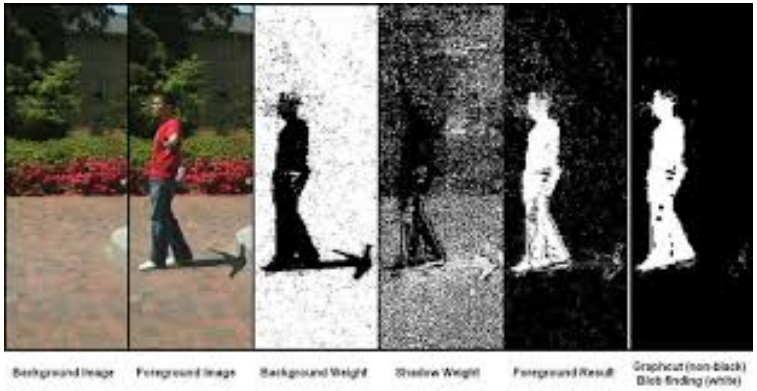
Energy optimization through performance

- Vectorization:



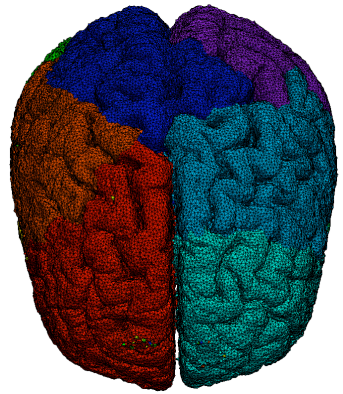
- up to 16x performance increase
- $\uparrow$ Performance $\Longrightarrow$ $\downarrow$ Energy
- compiler often can not identify a context to apply automatic vectorization

Background Subtraction(ECIT)



- Background Subtraction is a computer vision technique which is usually used to detect movement
- Implementation provided by ECIT(Fabian Campbell-West)
- Results of profiling:
  Only 10 % of energy is spent on the algorithm
  90 % of energy is spent on OpenCV
- Optimization through compiler:
  problems:auto-vectorization was not applied in `normL2Sqr`
  optimization strategy:align and restrict pointers,
  forced unroll,enable AVX-256
  results: 23 % reduction in energy consumption

Parallel speculative image-based mesh generation(CRTC)



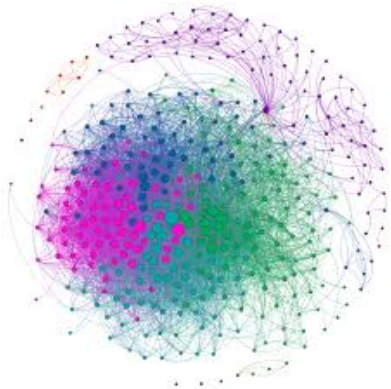
- Image-based meshing is the automated process of creating computer models for computational fluid dynamics (CFD) and finite element analysis (FEA) from 3D image data (such as magnetic resonance imaging (MRI), computed tomography (CT) or microtomography)
- Implementation provided by CRTC(Center for Real-Time Computing)

  problems: AVX-256 and auto-vectorization were not applied

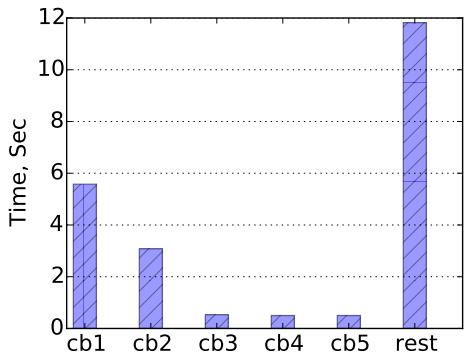
  optimization strategy:align and restrict pointers,
  forced unroll,enable AVX-256

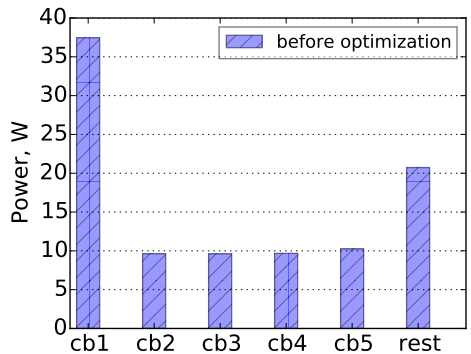
  results: 10 % reduction in energy consumption

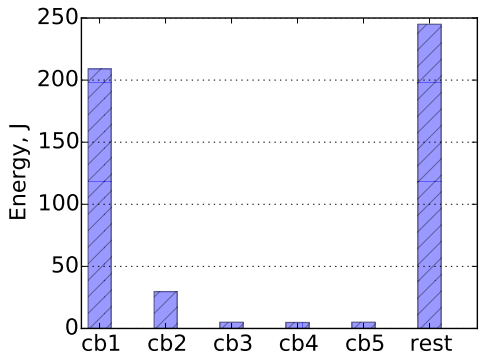
K-means energy



- K-means is the most popular algorithm for cluster analysis in data mining which is widely used in data centers
- Results of profiling:







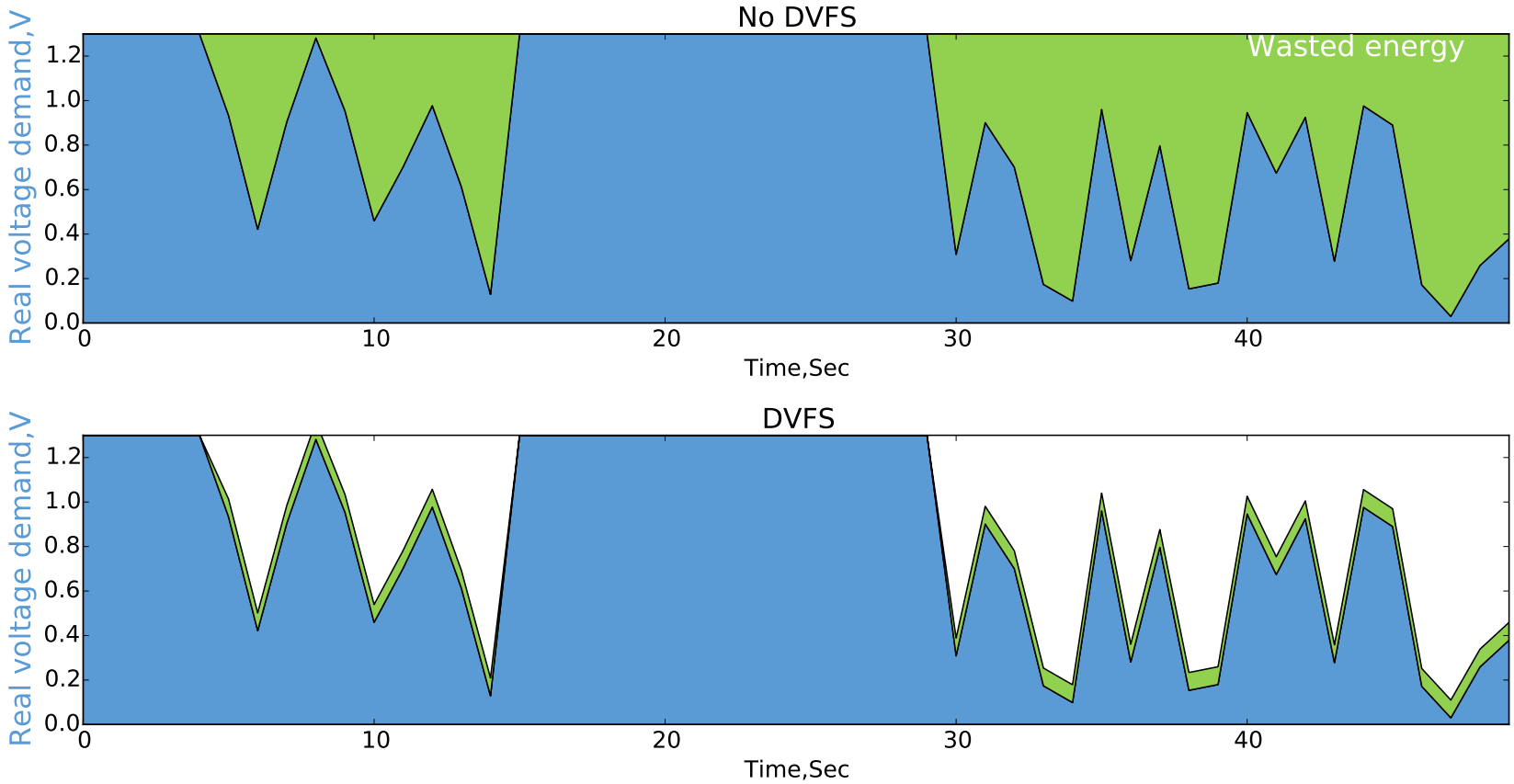
- Optimization through compiler:

  problems:unroll and auto-vectorization are not applied

  optimization strategy:align and to restrict pointers,forced unroll

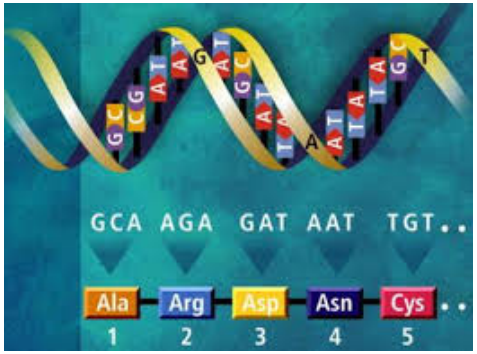
  results: 7x energy decrease

Energy optimization through DVFS

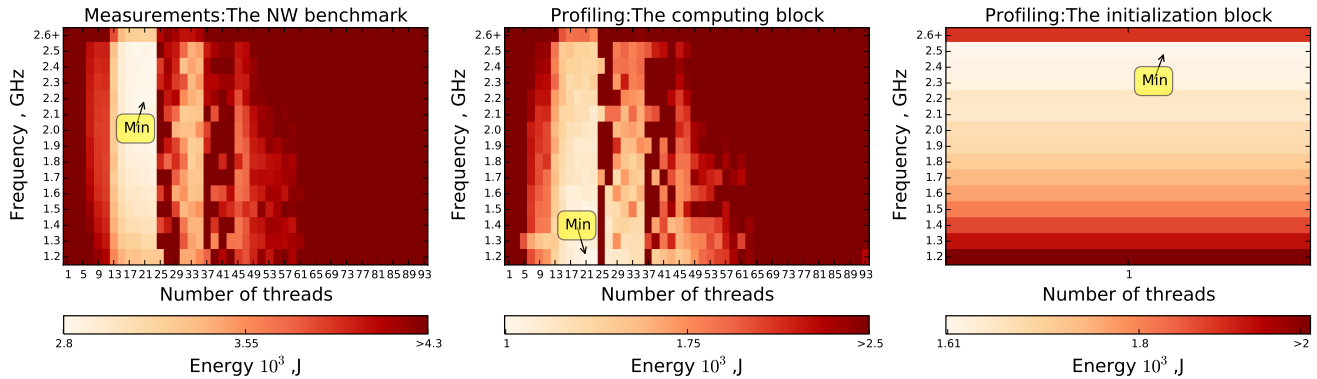


- Real voltage demand is than the supplied voltage
- Dynamic $\uparrow\downarrow$ Voltage $\Longrightarrow$ $\downarrow$ Energy

Needleman Wunsch algorithm



- The Needleman Wunsch algorithm is an algorithm used in bioinformatics to align protein or nucleotide sequences
- Results of profiling:



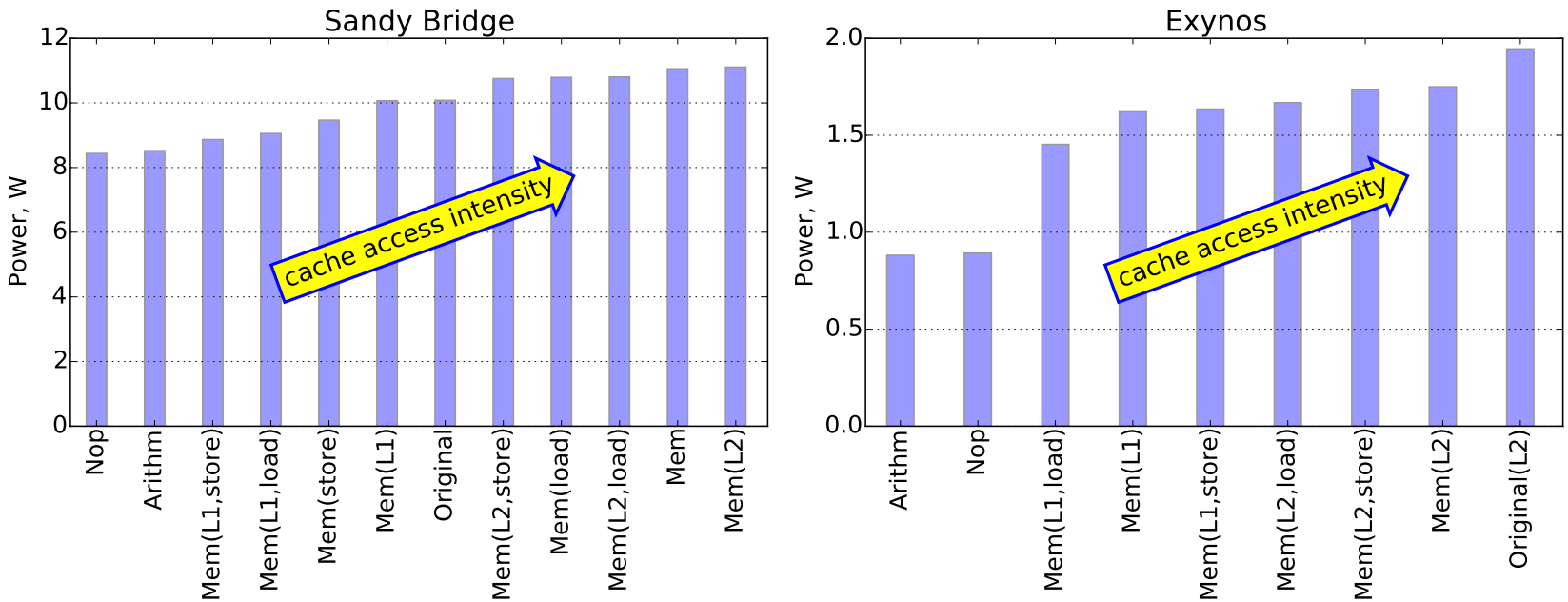
- Optimization strategy: DVFS

  results: 9 % energy reduction

  results: 41.6% peak power reduction

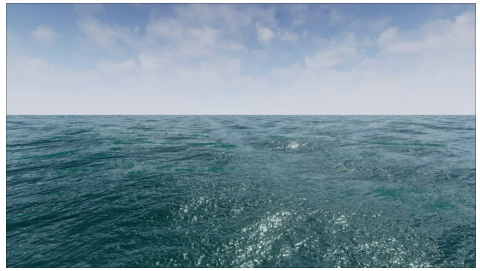
Power

Impact of cache access instructions

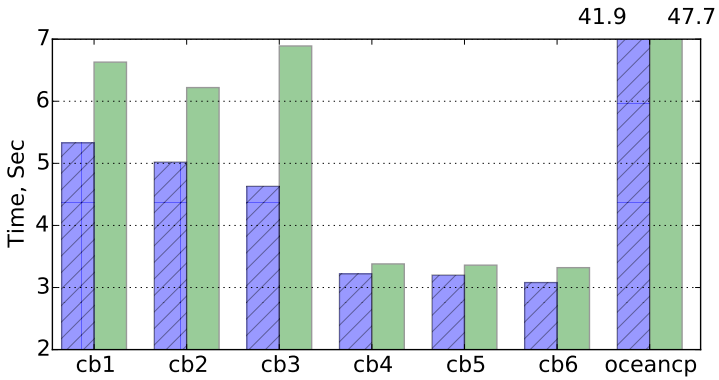


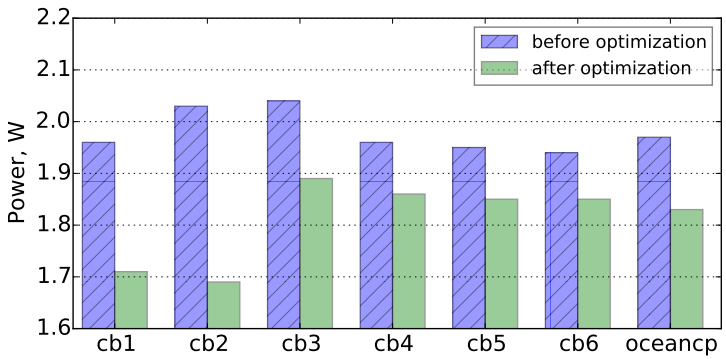
- CPU power is primarily affected by cache accesses

Ocean-cp power optimization



- Ocean-cp is a program that simulates large-scale ocean movements
- Results of profiling:



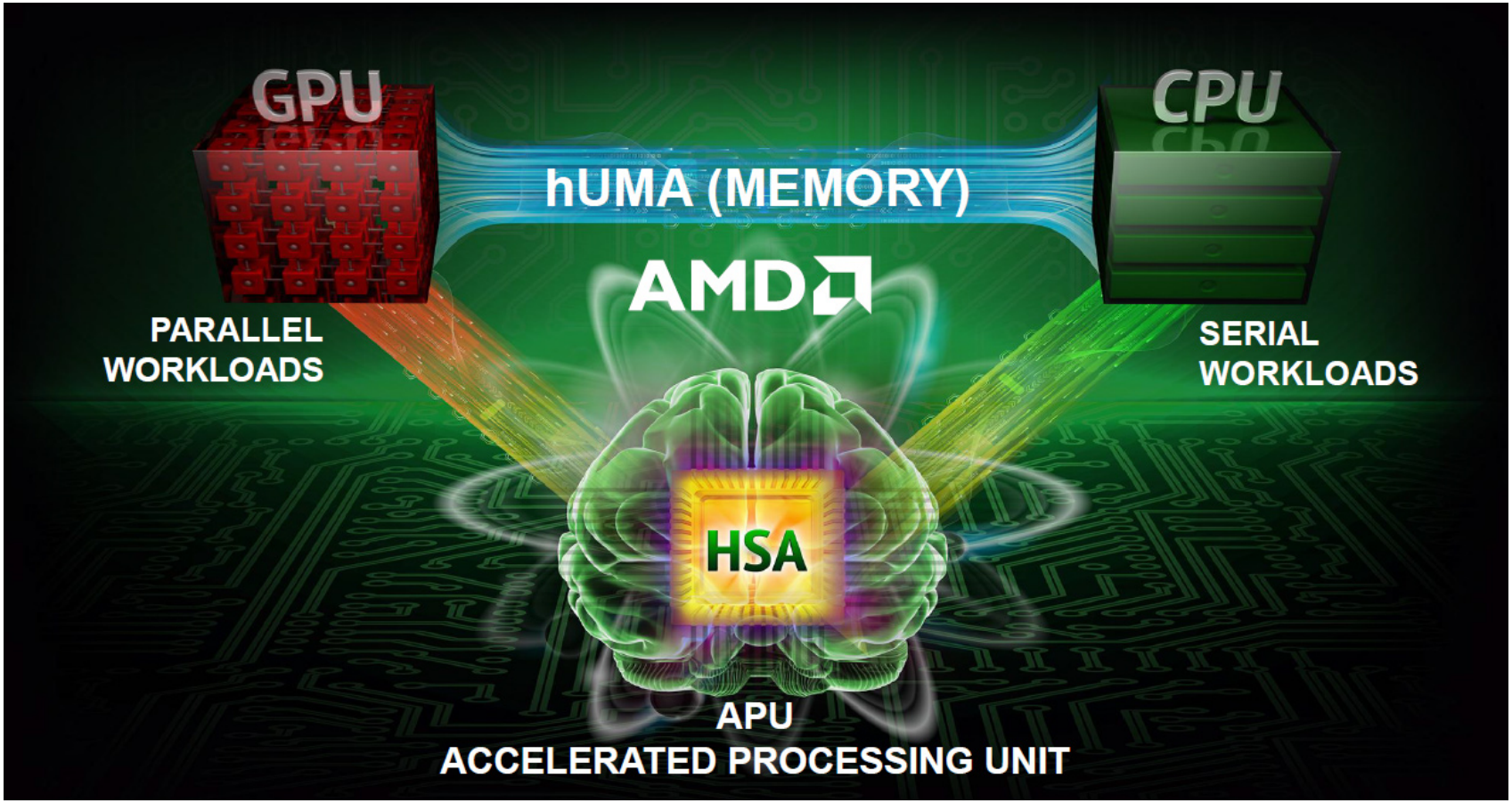


- Optimization through compiler:
  more than 50% of energy is spent on 6 blocks
  problems: redundant memory accesses
  optimization strategy: disable predictive commoning optimization
  results: 10 % power reduction

Outline

- Introduction
- Probabilistic approach
- Challenges
- ALEA implementation
- Use-cases
- **DIVIDEND and UNISERVER**

the DIVIDEND project



the UNISERVER project

applied micro®

Conclusion

- ALEA enables accurate energy accounting at the fine-grained level for many-core programming languages
- ALEA provides low overhead and architecture-independent approach
- ALEA could be effectively applied to optimize energy and power consumption
- Future work:
  fine-grained profiling at the instruction level
  port to new architectures(GPUs and Intel Xeon Phi)
  profiling of distributed systems

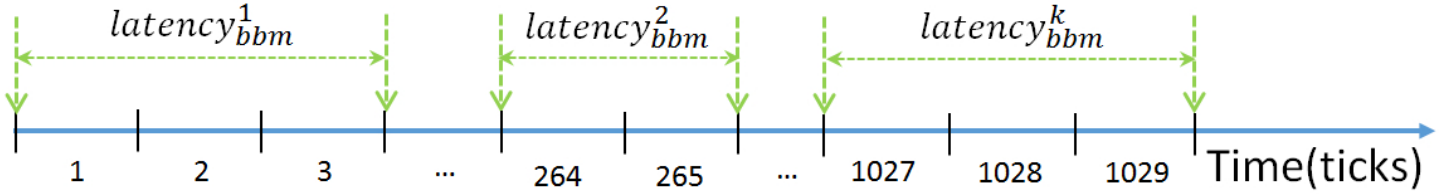
Thank you

This research has been supported by the UK EPSRC and by the EC FP7

BackUp

Probability to sample a basic block

- Basic block execution



- Introduce X_{bbm} associated with each tick:

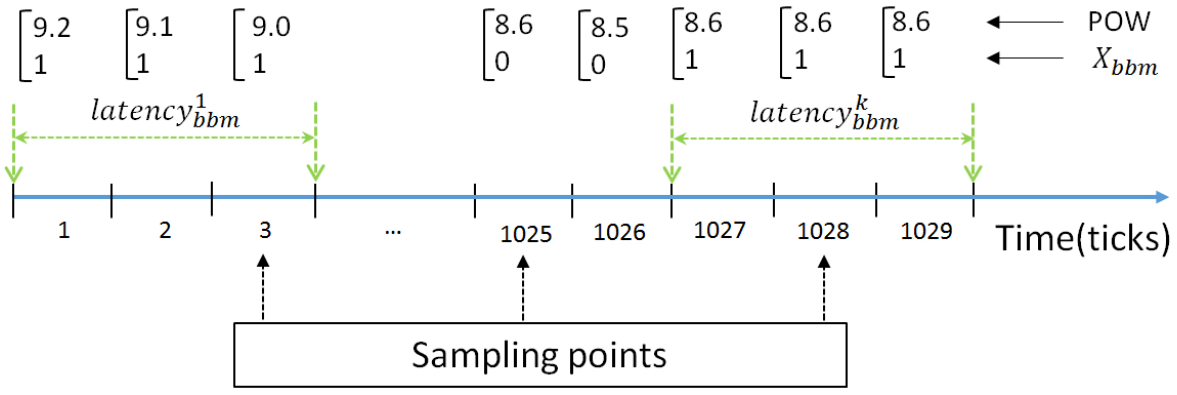
$$X_{bbm} = \begin{cases} 1, & \text{if } bbm \text{ is the sampled basic block} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

- Take one random sampling. Probability that *bbm* is sampled:

$$p_{bbm} = P(X_{bbm} = 1) = \frac{C^1_{t_{bbm}}}{C^1_{t_{exec}}} = \frac{\sum_{j=1}^{k} latency^j_{bbm}}{t_{exec}} = \frac{t_{bbm}}{t_{exec}} \tag{6}$$

Execution time estimates

- Take samples several times.Random sampling



- X_{bbm} random and follows the Bernoulli distribution
- Estimate p_{bbm} using the maximum likelihood estimator of parameter p_{bbm} in the Bernoulli distribution for X_{bbm}

$$\hat{p}_{bbm} = \frac{n_{bbm}}{n} \tag{7}$$

- t_{bbm} is estimated as

$$\hat{t}_{bbm} = \hat{p}_{bbm} \cdot t_{exec} = \frac{n_{bbm} \cdot t_{exec}}{n} \tag{8}$$

Power and Energy estimates

- The same probabilistic approach
- Power consumption is random variable(Normal distribution)
- Implementation of the variable is associated with each tick
- The mean power consumption of *bbm*:

$$\hat{pow}_{bbm} = \frac{1}{n_{bbm}} \cdot \sum_{i=1}^{n_{bbm}} pow_{bbm}^{i} \tag{9}$$

- Energy consumption of *bbm*:

$$\hat{e}_{bbm} = \hat{pow}_{bbm} \cdot \hat{t}_{bbm} \tag{10}$$

Quality of time estimates

- Confidence interval for p_{bbm}

$$\hat{p}^{u}_{bbm} = \hat{p}_{bbm} + z_{\alpha/2}\sqrt{\frac{1}{n} \cdot \hat{p}_{bbm} \cdot (1 - \hat{p}_{bbm})} \tag{11}$$

$$\hat{p}^{l}_{bbm} = \hat{p}_{bbm} - z_{\alpha/2}\sqrt{\frac{1}{n} \cdot \hat{p}_{bbm} \cdot (1 - \hat{p}_{bbm})} \tag{12}$$

$$\hat{p}^{l}_{bbm} \leq p \leq \hat{p}^{u}_{bbm} \tag{13}$$

- Confidence interval for t_{bbm}

$$\hat{p}^{l}_{bbm} \cdot t_{exec} \leq t_{bbm} \leq \hat{p}^{u}_{bbm} \cdot t_{exec} \tag{14}$$

Bounds and Confidence.Energy

- We can similarly build a confidence interval for power

$$\hat{pow}^{u}_{bbm} = \hat{pow}_{bbm} + z_{\alpha/2}\frac{s}{\sqrt{n_{bbm}}} \tag{15}$$

$$\hat{pow}^{l}_{bbm} = \hat{pow}_{bbm} - z_{\alpha/2}\frac{s}{\sqrt{n_{bbm}}} \tag{16}$$

$$s = \sqrt{\frac{1}{n_{bbm}-1}\cdot\sum_{i=1}^{n_{bbm}}(pow^{i}_{bbm} - \hat{pow}_{bbm})^2} \tag{17}$$

$$\hat{pow}^{l}_{bbm} \leq pow_{bbm} \leq \hat{pow}^{u}_{bbm} \tag{18}$$

- Confidence interval for energy consumption

$$\hat{p}^{l}_{bbm}\cdot t_{exec}\cdot\hat{pow}^{l}_{bbm} \leq e_{bbm} \leq \hat{p}^{u}_{bbm}\cdot t_{exec}\cdot\hat{pow}^{u}_{bbm} \tag{19}$$

Parallel applications

- **Basic block vector(BBV)** $\vec{bbm}$

$$\vec{bbm} = bb_{thread_1}, bb_{thread_2}, ..., bb_{thread_l} \tag{20}$$

$$\hat{t}_{\vec{bbm}} = \hat{p}_{\vec{bbm}} \cdot t_{exec} = \frac{n_{\vec{bbm}} \cdot t_{exec}}{n} \tag{21}$$

$$\hat{pow}_{\vec{bbm}} = \frac{1}{n_{\vec{bbm}}} \cdot \sum_{i=1}^{n_{\vec{bbm}}} pow^{i}_{\vec{bbm}} \tag{22}$$

$$\hat{e}_{\vec{bbm}} = \hat{pow}_{\vec{bbm}} \cdot \hat{t}_{\vec{bbm}} \tag{23}$$

Platforms

Intel Sandy Bridge
(Intel Xeon E5-2650), 2 CPU, 8 cores, 32KB/32KB I/D-Cache per core, 2MB L2 cache, 20MB L3 cache. OS: CentOS (release 6.5). Frequency: 2 GHz.
Energy measurements: RAPL

Samsung Exynos 5 Octa(Odroid-XU+E),
ARM Big.LITTLE, 4 A15 cores,4
A7 cores, 32KB/32KB I/D-Cache per core, 2MB L2 cache, OS: Ubuntu 14.04 LTS.Frequency:1.6 GHz
Energy measurements: Power meters(INA 231)