Energy-performance trade-offs for HPC applications on low power and high-end systems

E. Calore A. Gabbana S. F. Schifano R. Tripiccione

INFN Ferrara and Università degli Studi di Ferrara, Italy

COSA Meeting

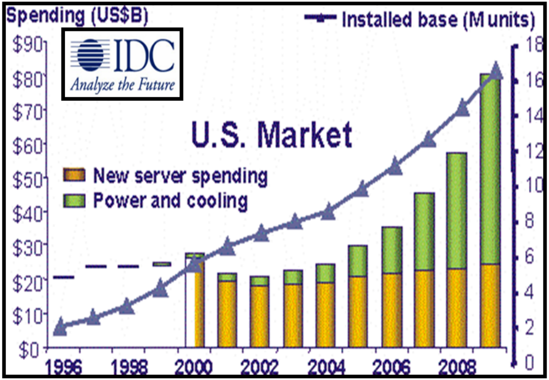
Bologna, November 3rd 2016

Outline

Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
    - NVIDIA Jetson TK1
    - 96Boards HiKey
    - Results
5. High-End Pocessors
    - Intel Xeon E5-2630v3 Haswell
    - NVIDIA K80 (half)
    - Results
6. Actual full HPC node
    - Function-by-function tuning
    - Results
7. Conclusions

Energy is becoming more and more important in HPC



HPC facilities may start to account for consumed energy instead of running time

Two research approaches...

Use low-power/embedded hardware for HPC

- may consume less since hardware is designed to be low-power
- may also cost less thanks to economy of scale

Minimize energy consumption on actual high-end systems

- may be possible using new energy monitoring / control hardware
- may be possible by software optimization / tuning

Two research approaches...

Use low-power/embedded hardware for HPC

- may consume less since hardware is designed to be low-power
- may also cost less thanks to economy of scale

Minimize energy consumption on actual high-end systems

- may be possible using new energy monitoring / control hardware
- may be possible by software optimization / tuning

Outline

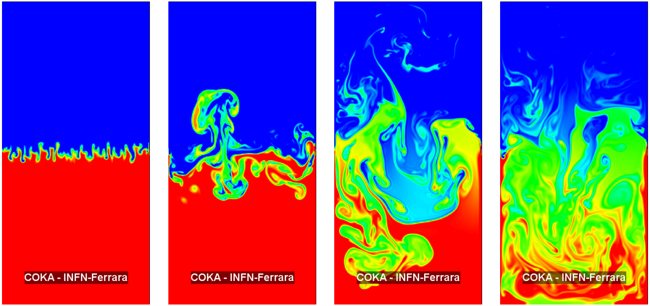
1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
    - NVIDIA Jetson TK1
    - 96Boards HiKey
    - Results
5. High-End Pocessors
    - Intel Xeon E5-2630v3 Haswell
    - NVIDIA K80 (half)
    - Results
6. Actual full HPC node
    - Function-by-function tuning
    - Results
7. Conclusions

The D2Q37 Lattice Boltzmann Model

- Lattice Boltzmann method (LBM) is a class of computational fluid dynamics (CFD) methods
- LBM methods simulate a discrete **Boltzmann** equation, which under certain conditions, reduce to the **Navier-Stokes** equation
- **virtual particles** called **populations** arranged at edges of a discrete and regular grid are used to simulate a synthetic and simplified dynamics
- the interaction is implemented by two main functions applied to the virtual particles: **propagation** and **collision**
- D2Q37 is a D2 model with 37 components of velocity (populations)
- suitable to study behaviour of **compressible** gas and fluids optionally in presence of **combustion** effects
- correct treatment of Navier-Stokes, heat transport and perfect-gas ($P = \rho T$) equations

Simulation of the Rayleigh-Taylor (RT) Instability

Instability at the interface of two fluids of different densities triggered by gravity.



A cold-dense fluid over a less dense and warmer fluid triggers an instability that mixes the two fluid-regions (till equilibrium is reached).

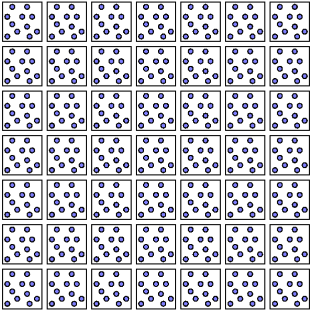
Computational Scheme of LBM

```
foreach time−step

  foreach lattice−point
    propagate();
  endfor

  foreach lattice−point
    collide();
  endfor

endfor
```



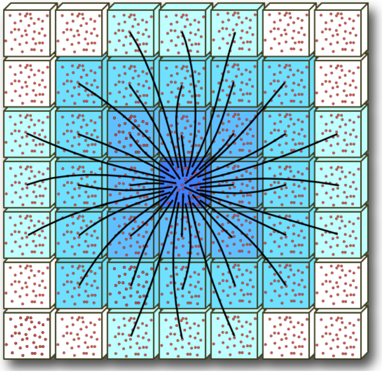
Embarassing parallelism

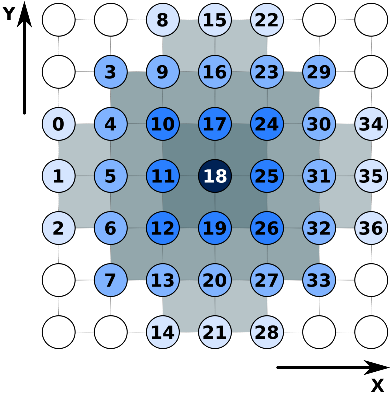
All sites can be processed in parallel applying in sequence propagate and collide.

Challenge

Design an efficient implementation able exploit a large fraction of available peak performance.

D2Q37: propagation scheme





- perform accesses to neighbour-cells at distance 1,2, and 3
- generate memory-accesses with **sparse** addressing patterns

D2Q37 collision

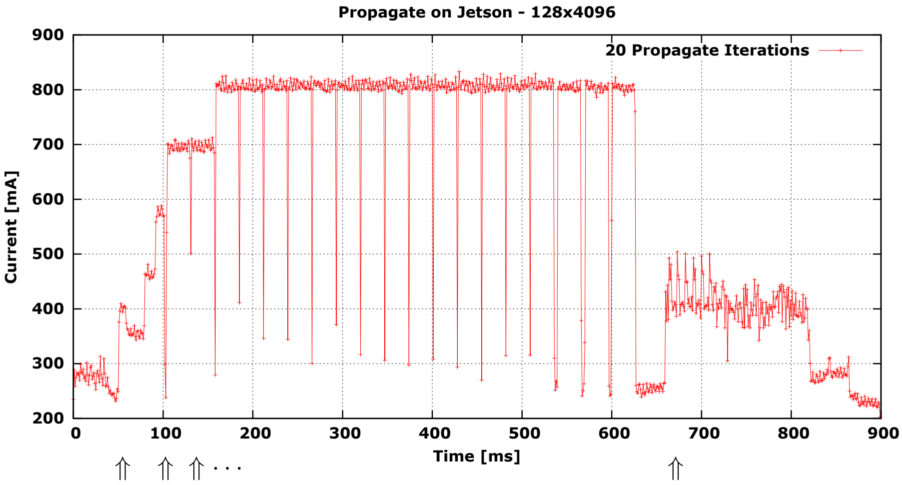
- collision is computed at each lattice-cell after computation of boundary conditions
- computational intensive: for the D2Q37 model requires $\approx$ 7500 DP floating-point operations
- completely local: arithmetic operations require only the populations associate to the site

- computation of propagate and collide kernels are kept separate
- after propagate but before collide we may need to perform collective operations (e.g. divergence of of the velocity field) if we include computations conbustion effects.

Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
   - NVIDIA Jetson TK1
   - 96Boards HiKey
   - Results
5. High-End Pocessors
   - Intel Xeon E5-2630v3 Haswell
   - NVIDIA K80 (half)
   - Results
6. Actual full HPC node
   - Function-by-function tuning
   - Results
7. Conclusions

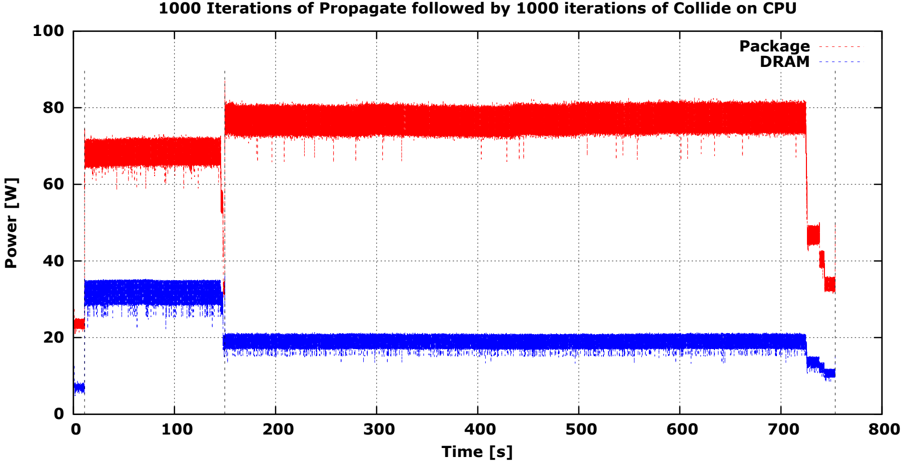
Acquired data example with default frequency scaling



Iterations can be counted

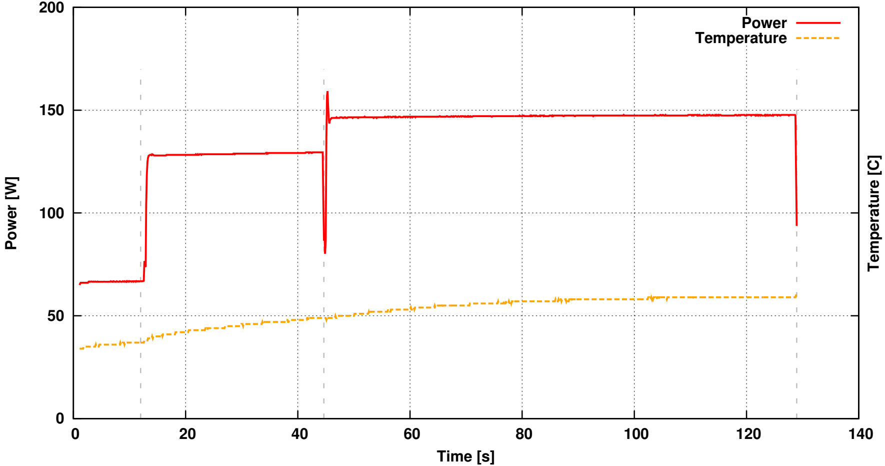
This is a D2H transfer

Acquired data example using RAPL counters



Intel Haswell CPU RAPL counters acquired at 100Hz and converted in Watt; acquisition performed with a custom developed wrapper to the PAPI library. Lattice: 1024 × 8192. Requested CPU clock: 2.4GHz.

Acquired data example using NVML



Half of an NVIDIA K80 GPU. Acquisition performed with a custom developed wrapper to the PAPI library. Lattice: 1024 × 8192. Requested GPU clock: 875MHz.

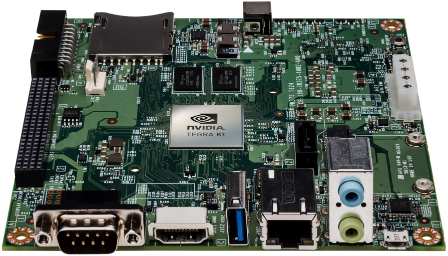
Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
    - NVIDIA Jetson TK1
    - 96Boards HiKey
    - Results
5. High-End Pocessors
    - Intel Xeon E5-2630v3 Haswell
    - NVIDIA K80 (half)
    - Results
6. Actual full HPC node
    - Function-by-function tuning
    - Results
7. Conclusions

Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
   - NVIDIA Jetson TK1
   - 96Boards HiKey
   - Results
5. High-End Pocessors
   - Intel Xeon E5-2630v3 Haswell
   - NVIDIA K80 (half)
   - Results
6. Actual full HPC node
   - Function-by-function tuning
   - Results
7. Conclusions

NVIDIA Jetson TK1



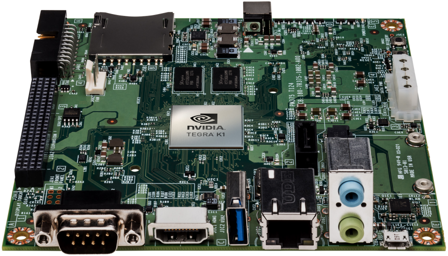
SoC: Tegra K1

- CPU: NVIDIA "4-Plus-1" 2.32GHz ARM quad-core Cortex-A15, with battery-saving shadow-core
- GPU: NVIDIA Kepler "GK20a" GPU with 192 SM3.2 CUDA cores

Awarded for the Best Paper

7th Workshop on UnConventional High Performance Computing (UCHPC), Porto 2014

NVIDIA Jetson TK1



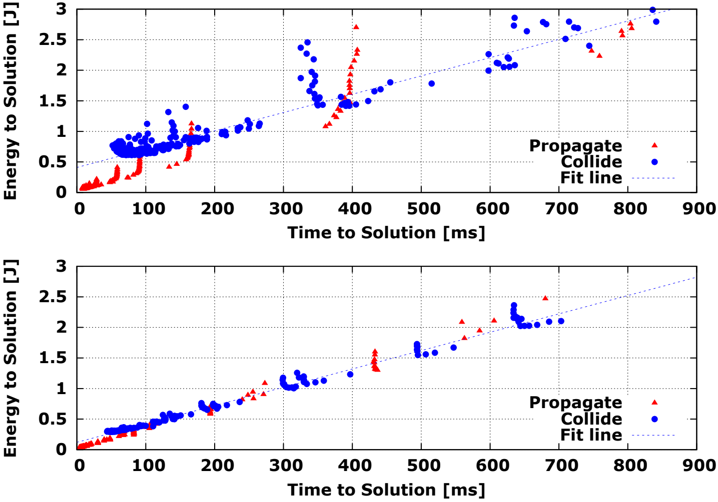
SoC: Tegra K1

- CPU: NVIDIA "4-Plus-1" 2.32GHz ARM quad-core Cortex-A15, with battery-saving shadow-core
- GPU: NVIDIA Kepler "GK20a" GPU with 192 SM3.2 CUDA cores

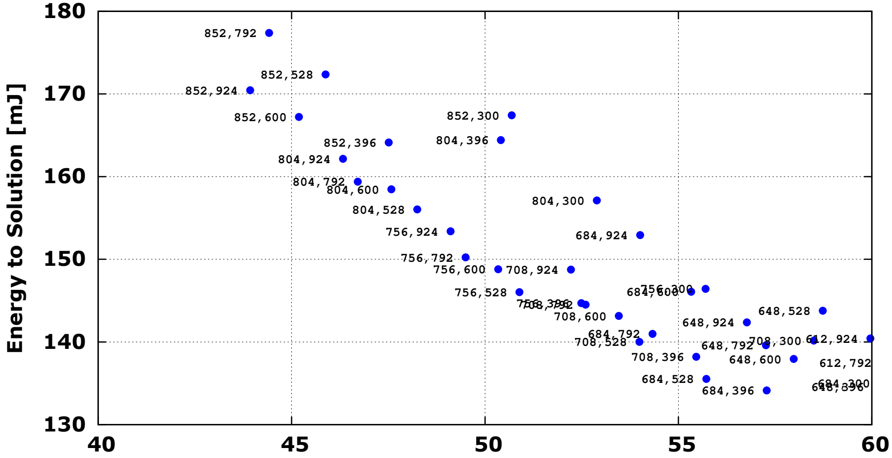
Awarded for the Best Paper

7th Workshop on UnConventional High Performance Computing (UCHPC), Porto 2014

Energy to Sol. vs Time to Sol. CPU(top), GPU(bottom)



Energy to Solution vs Time to Solution (GPU GK20A) zoom

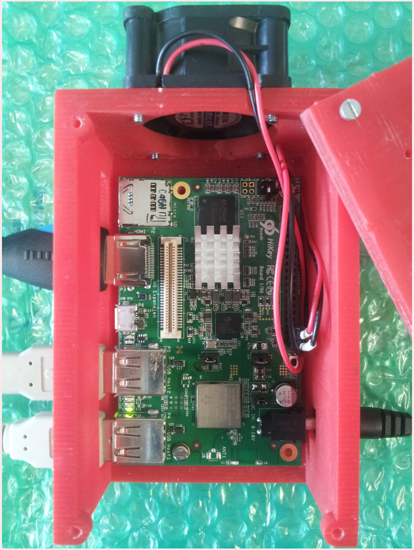


(E_s computed after removing the idle power drain)

Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
   - NVIDIA Jetson TK1
   - 96Boards HiKey
   - Results
5. High-End Processors
   - Intel Xeon E5-2630v3 Haswell
   - NVIDIA K80 (half)
   - Results
6. Actual full HPC node
   - Function-by-function tuning
   - Results
7. Conclusions

96Boards - HiKey



SoC: HiSilicon Kirin 6220

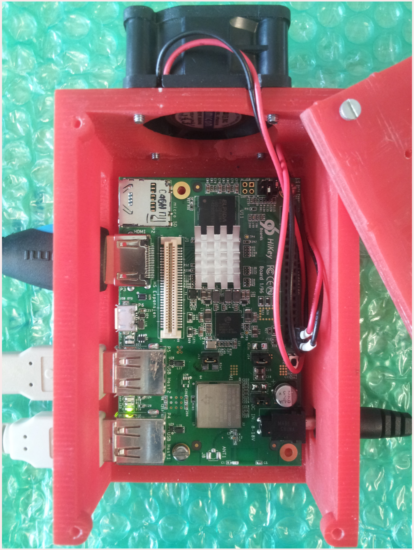
- CPU: 8 core ARM Cortex-A53 running at 1.2GHz (64-bit aarch64)
- GPU: ARM Mali 450-MP4 GPU
- MEM: 1GB of 800MHz LPDDR3

Awarded for the Best Paper

8^{th} Workshop on UnConventional High Performance Computing (UCHPC), Vienna 2015

3D printed case to fit a fan
(Thanks to V. Carassiti and A. Cotta Ramusino, INFN Ferrara)

96Boards - HiKey



SoC: HiSilicon Kirin 6220

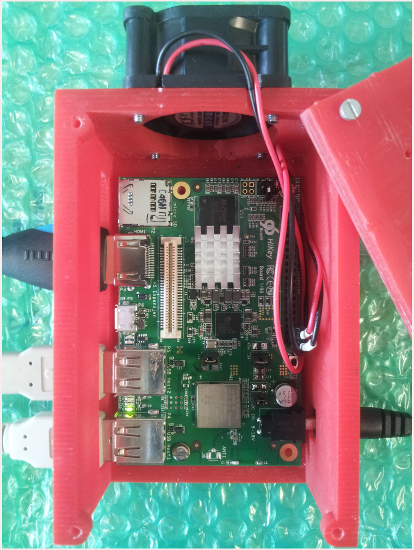
- CPU: 8 core ARM Cortex-A53 running at 1.2GHz (64-bit aarch64)
- GPU: ARM Mali 450-MP4 GPU
- MEM: 1GB of 800MHz LPDDR3

Awarded for the Best Paper

8th Workshop on UnConventional High Performance Computing (UCHPC), Vienna 2015

3D printed case to fit a fan (Thanks to V. Carassiti and A. Cotta Ramusino, INFN Ferrara)

96Boards - HiKey



SoC: HiSilicon Kirin 6220

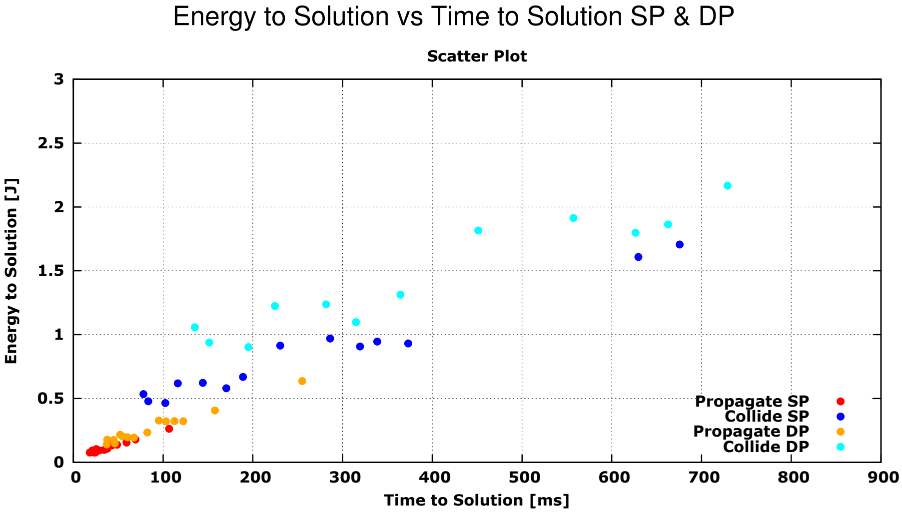
- CPU: 8 core ARM Cortex-A53 running at 1.2GHz (64-bit aarch64)
- GPU: ARM Mali 450-MP4 GPU
- MEM: 1GB of 800MHz LPDDR3

Awarded for the Best Paper

8th Workshop on UnConventional High Performance Computing (UCHPC), Vienna 2015

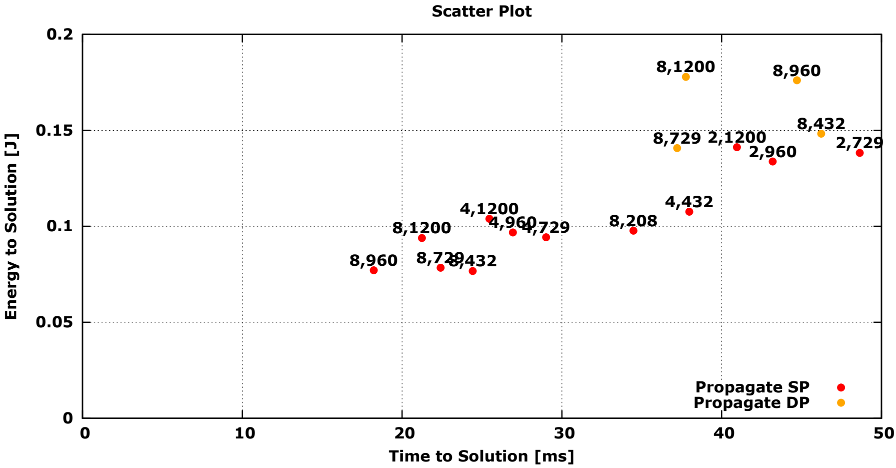
3D printed case to fit a fan (Thanks to V. Carassiti and A. Cotta Ramusino, INFN Ferrara)

C with NEON intrinsics, on the Cortex A53

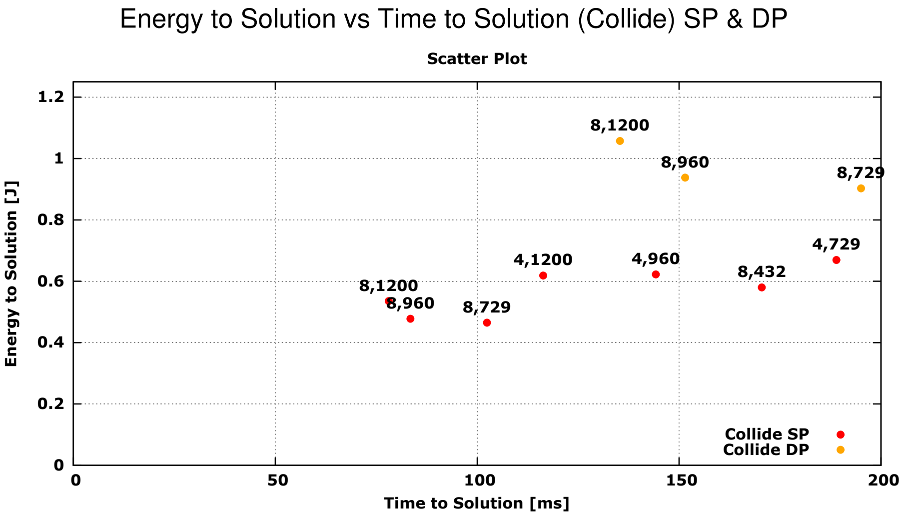


C with NEON intrinsics, on the Cortex A53

Energy to Solution vs Time to Solution (Propagate) SP & DP



C with NEON intrinsics, on the Cortex A53



Outline

Preliminary conclusions about Low-Power Processors

Conclusions

- limited but not negligible power optimization is possible by adjusting clocks on a kernel-by-kernel basis (between $\approx 5 \cdots 25\%$).
- baseline power consumption is relevant ($\approx 30\%$)
- hard to differentiate between leakage current and ancillary electronics
- options to run the processor at very low frequencies seem almost useless (at least for the adopted benchmark)

Processor	E_S [J] per iter.	T_S [ms] per iter.	EDP [J s]
GK20A	0.30	42	0.013
ARM A15	0.67	58	0.039
ARM A53	0.52	77	0.040

Table: Best EDP values, with corresponding *energy-to-solution* and *time-to-solution*, running the (SP) collide kernel. Lattice 128x1024

Preliminary conclusions about Low-Power Processors

Conclusions

- limited but not negligible power optimization is possible by adjusting clocks on a kernel-by-kernel basis (between $\approx 5 \cdots 25\%$).
- baseline power consumption is relevant ($\approx 30\%$)
- hard to differentiate between leakage current and ancillary electronics
- options to run the processor at very low frequencies seem almost useless (at least for the adopted benchmark)

Processor	E_S [J] per iter.	T_S [ms] per iter.	EDP [J s]
GK20A	0.30	42	0.013
ARM A15	0.67	58	0.039
ARM A53	0.52	77	0.040

Table: Best EDP values, with corresponding *energy-to-solution* and *time-to-solution*, running the (SP) collide kernel. Lattice 128x1024

Outline

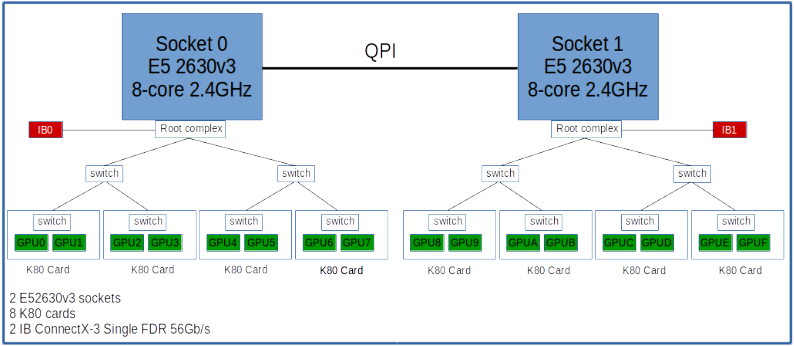
COKA Cluster Overview

Compute nodes:

Supermicro SYS-4028GR-TR

- 2 x Intel Xeon E5-2630v3
- 8 x NVIDIA K80 (2xGPU)
- 2 x Mellanox ConnectX-3 Single FDR 56Gb/s Infiniband cards

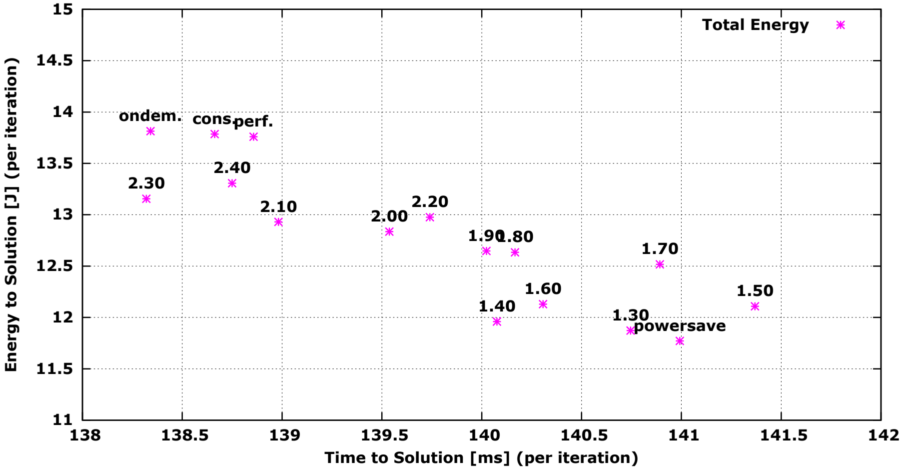
Nodes Overview



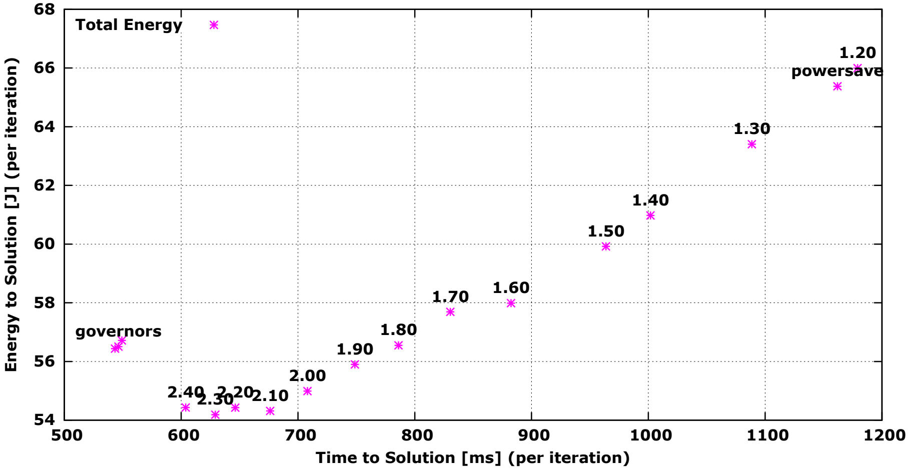
2 E52630v3 sockets
8 K80 cards
2 IB ConnectX-3 Single FDR 56Gb/s

Outline

Energy/Time to Solution Propagate DP



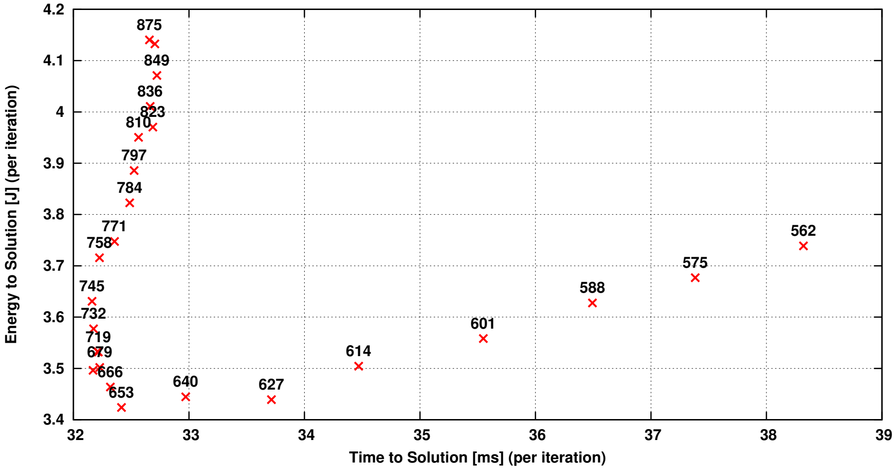
Energy/Time to Solution Collide DP



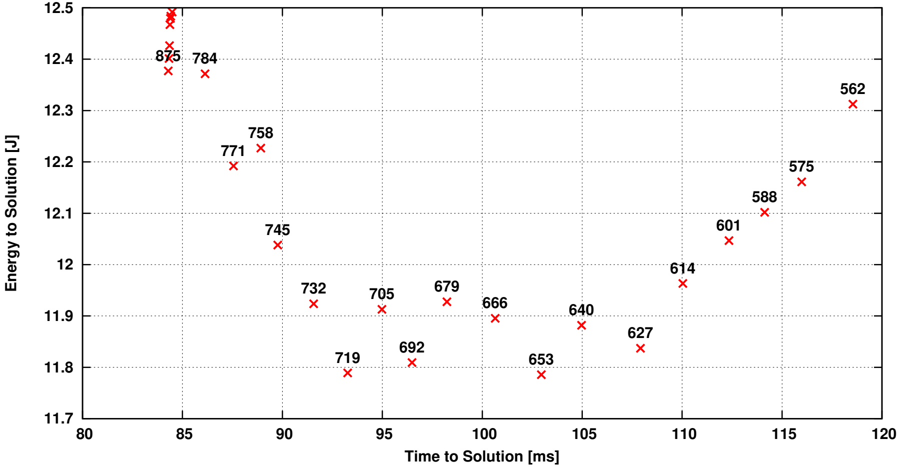
Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
    - NVIDIA Jetson TK1
    - 96Boards HiKey
    - Results
5. High-End Pocessors
    - Intel Xeon E5-2630v3 Haswell
    - **NVIDIA K80 (half)**
    - Results
6. Actual full HPC node
    - Function-by-function tuning
    - Results
7. Conclusions

Energy/Time to Solution Propagate DP



Energy/Time to Solution Collide DP



Outline

Results for single processors

Taking into account, for both CPU and GPU processors, the frequencies that led the best energy efficiency, we estimated the energy saving wrt the performance penalty:

	GPU		**CPU**	
	E_S saving	T_S cost	E_S saving	T_S cost
propagate	18%	0%	9%	3%
collide	6%	10%	4%	4%
Full code	11%	10%	7%	8%

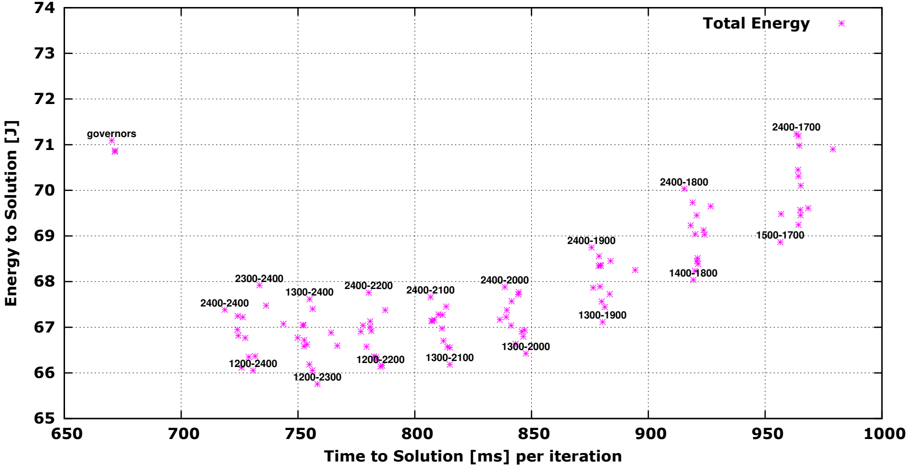
Table: Energy-to-solution (E_S) gains and the corresponding time-to-solution (T_S) costs.

Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
    - NVIDIA Jetson TK1
    - 96Boards HiKey
    - Results
5. High-End Pocessors
    - Intel Xeon E5-2630v3 Haswell
    - NVIDIA K80 (half)
    - Results
6. **Actual full HPC node**
    - Function-by-function tuning
    - Results
7. Conclusions

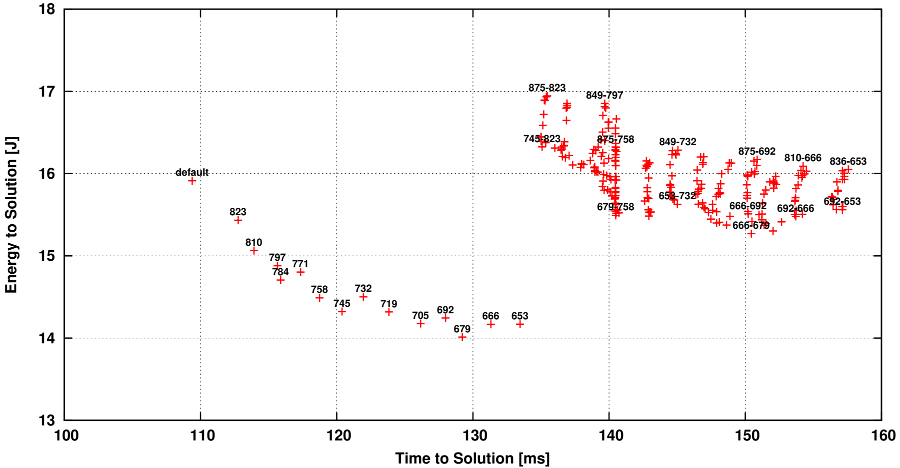
Outline

Function-by-function tuning on CPUs



Cost of $\approx 10\mu s$ for each frequency change.

Function-by-function tuning on GPUs



Cost of a frequency change is $\approx 10ms$, thus identifying a single GPU frequency for the whole simulation seems a better choice.

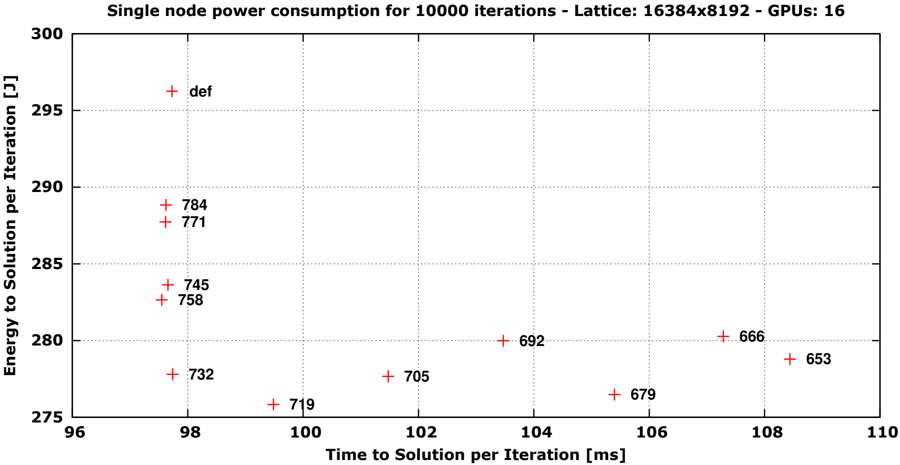
Outline

1. Introduction
2. A Lattice Boltzmann Model as a benchmark
3. Measuring the energy consumption
4. Low Power SoC
   - NVIDIA Jetson TK1
   - 96Boards HiKey
   - Results
5. High-End Pocessors
   - Intel Xeon E5-2630v3 Haswell
   - NVIDIA K80 (half)
   - Results
6. Actual full HPC node
   - Function-by-function tuning
   - Results
7. Conclusions

Power drained for GPUs at different fixed frequencies

Full production code running on 16 GPUs hosted on a single node.
Power drain measured from power supplies, through IPMI.

Energy consumption for GPUs at fixed frequencies



At a specific frequency (i.e. 732MHz) $\approx$ 7% of the total consumed energy of the computing node can be saved without impacting performances.

Outline

1 Introduction

2 A Lattice Boltzmann Model as a benchmark

3 Measuring the energy consumption

4 Low Power SoC
  - NVIDIA Jetson TK1
  - 96Boards HiKey
  - Results

5 High-End Pocessors
  - Intel Xeon E5-2630v3 Haswell
  - NVIDIA K80 (half)
  - Results

6 Actual full HPC node
  - Function-by-function tuning
  - Results

7 Conclusions

Conclusions

- default frequency governors do not seems to be energy aware;
- per function frequency optimization is not viable yet on GPUs, but it is on CPUs;
- per application frequency optimization can give interesting energy savings with minimal or no impact on performances on both CPU and GPUs;
- in general, for **compute bound** functions higher clocks are desiderable for both energy efficiency and performances, while for **memory bound** functions clocks can often be reduced to minimize energy consumption minimally impacting on performances;

Future works

- perform similar analisys on P100, KNL and other architectures
- collect data for a fair comparison between architectures for several metrics
- evaluate communication costs between different processors

Conclusions

- default frequency governors do not seems to be energy aware;
- per function frequency optimization is not viable yet on GPUs, but it is on CPUs;
- per application frequency optimization can give interesting energy savings with minimal or no impact on performances on both CPU and GPUs;
- in general, for **compute bound** functions higher clocks are desiderable for both energy efficiency and performances, while for **memory bound** functions clocks can often be reduced to minimize energy consumption minimally impacting on performances;

Future works

- perform similar analisys on P100, KNL and other architectures
- collect data for a fair comparison between architectures for several metrics
- evaluate communication costs between different processors

Thanks for Your attention