

MONT-BLANC

🌐 montblanc-project.eu | @MontBlanc_EU

Mobile technology for production-ready high-performance computing systems: The path of the Mont-Blanc project

Filippo Mantovani

October 23rd, 2017



Barcelona Supercomputing Center

BSC-CNS objectives:

- Supercomputing services to Spanish and EU researchers
- R&D in Computer, Life, Earth and Engineering Sciences
- PhD programme, technology transfer, public engagement

BSC-CNS is a consortium that includes:

Spanish Government

60%



Catalan Government

30%



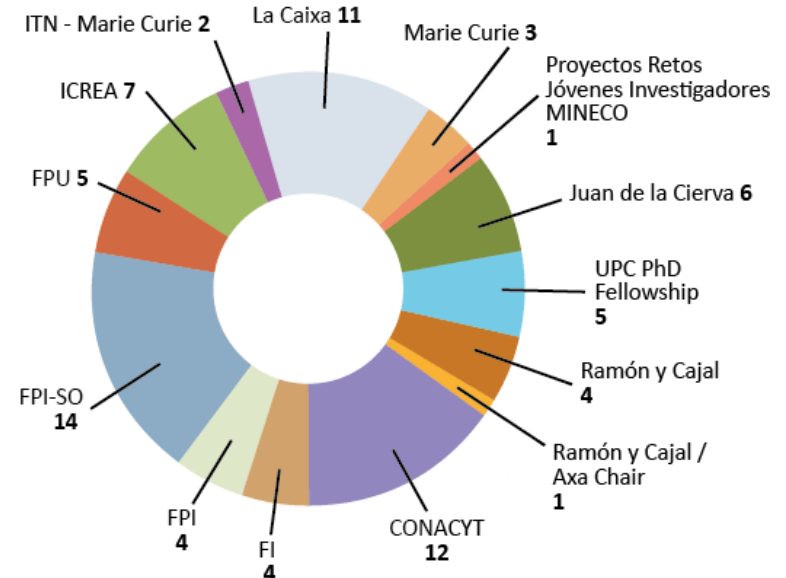
Univ. Politècnica de Catalunya (UPC)

10%



517 people from 44 countries * 31th of May 2017

Staff Funding (People): 517



Mission of BSC Scientific Departments



Computer Sciences

To influence the way machines are built, programmed and used: programming models, performance tools, Big Data, computer architecture, energy efficiency



Earth Sciences

To develop and implement global and regional state-of-the-art models for short-term air quality forecast and long-term climate applications



Life Sciences

To understand living organisms by means of theoretical and computational methods (molecular modeling, genomics, proteomics)

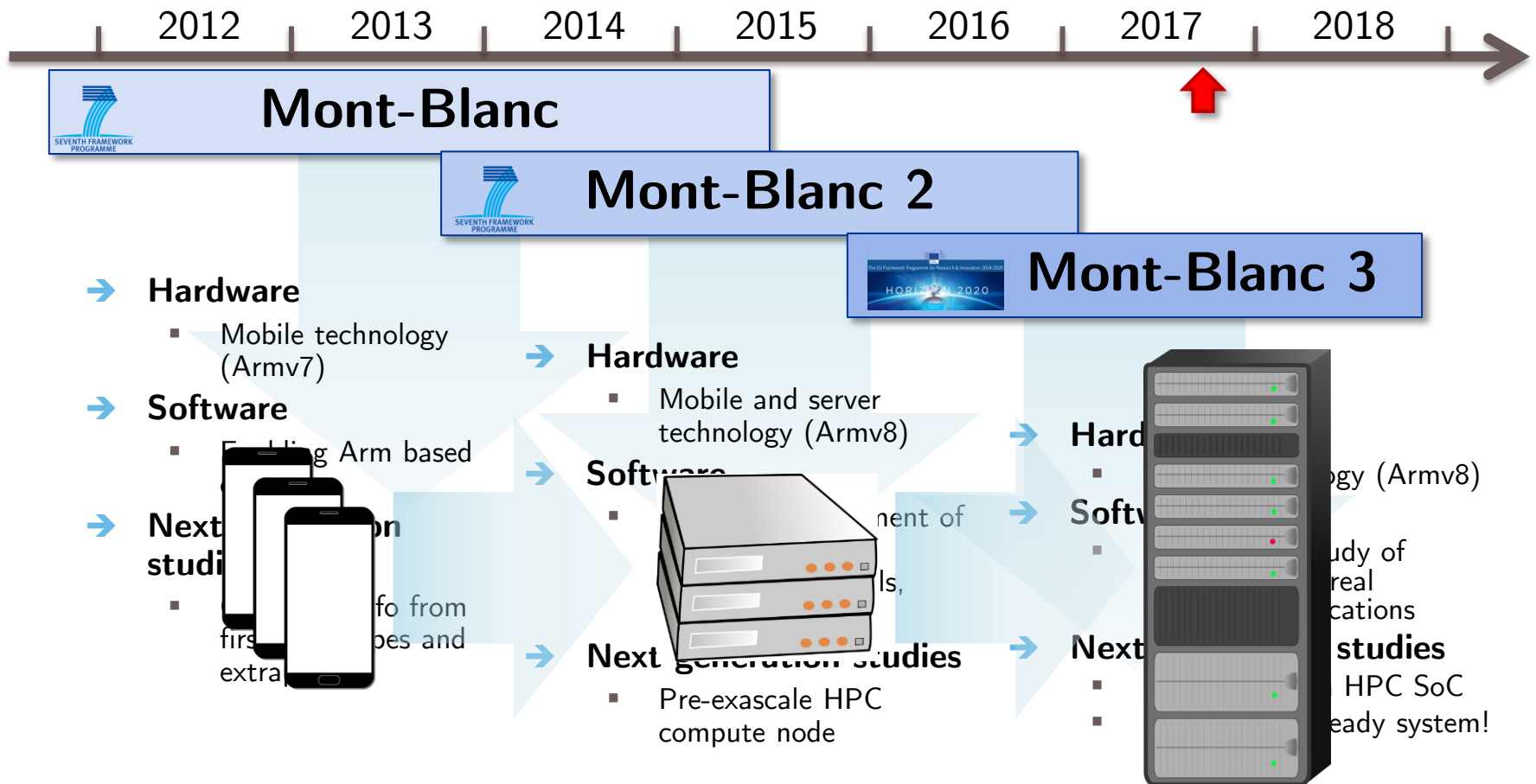


CASE

To develop scientific and engineering software to efficiently exploit super-computing capabilities (biomedical, geophysics, atmospheric, energy, social and economic simulations)

Mont-Blanc projects in a glance

Vision: to leverage the fast growing market of mobile technology for scientific computation, HPC and data centers.



Mont-Blanc contributions

Arm-based prototypes

- Mobile technology
- Server technology
- System integration



System software

- Programming model
- Performance analysis tools
- Evaluation of Arm-based ecosystem for HPC



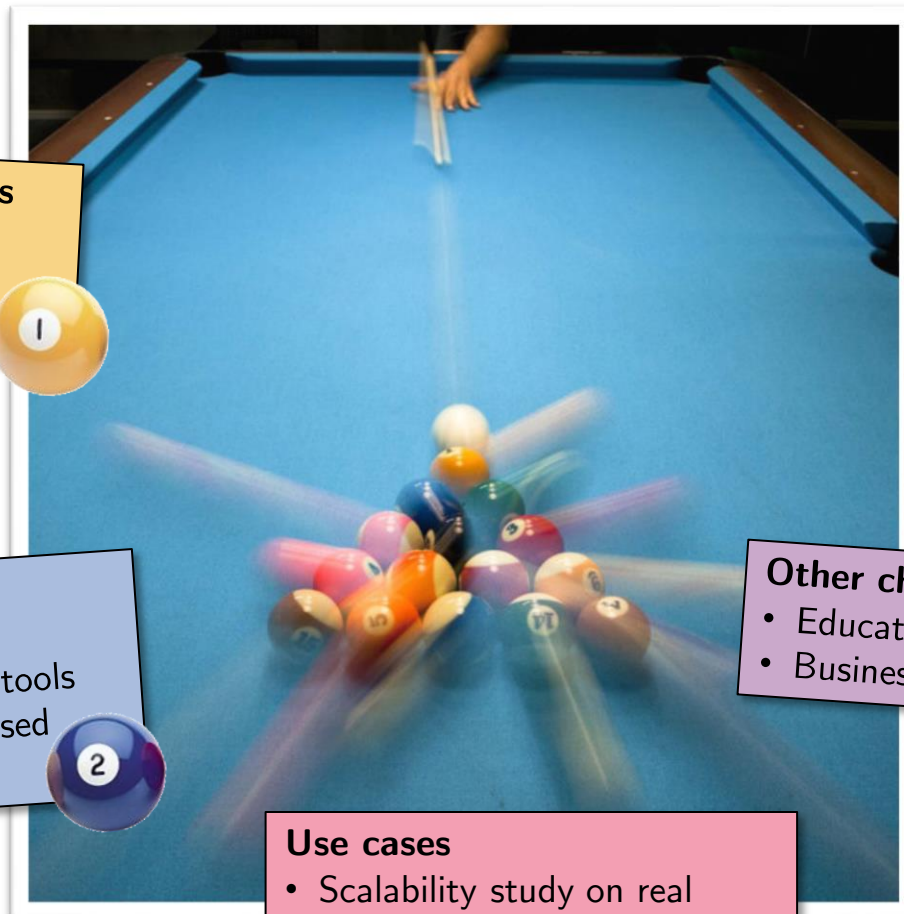
Use cases

- Scalability study on real Arm-based platforms
- Correlation of performance and power measurements
- Runtime features for future HPC systems



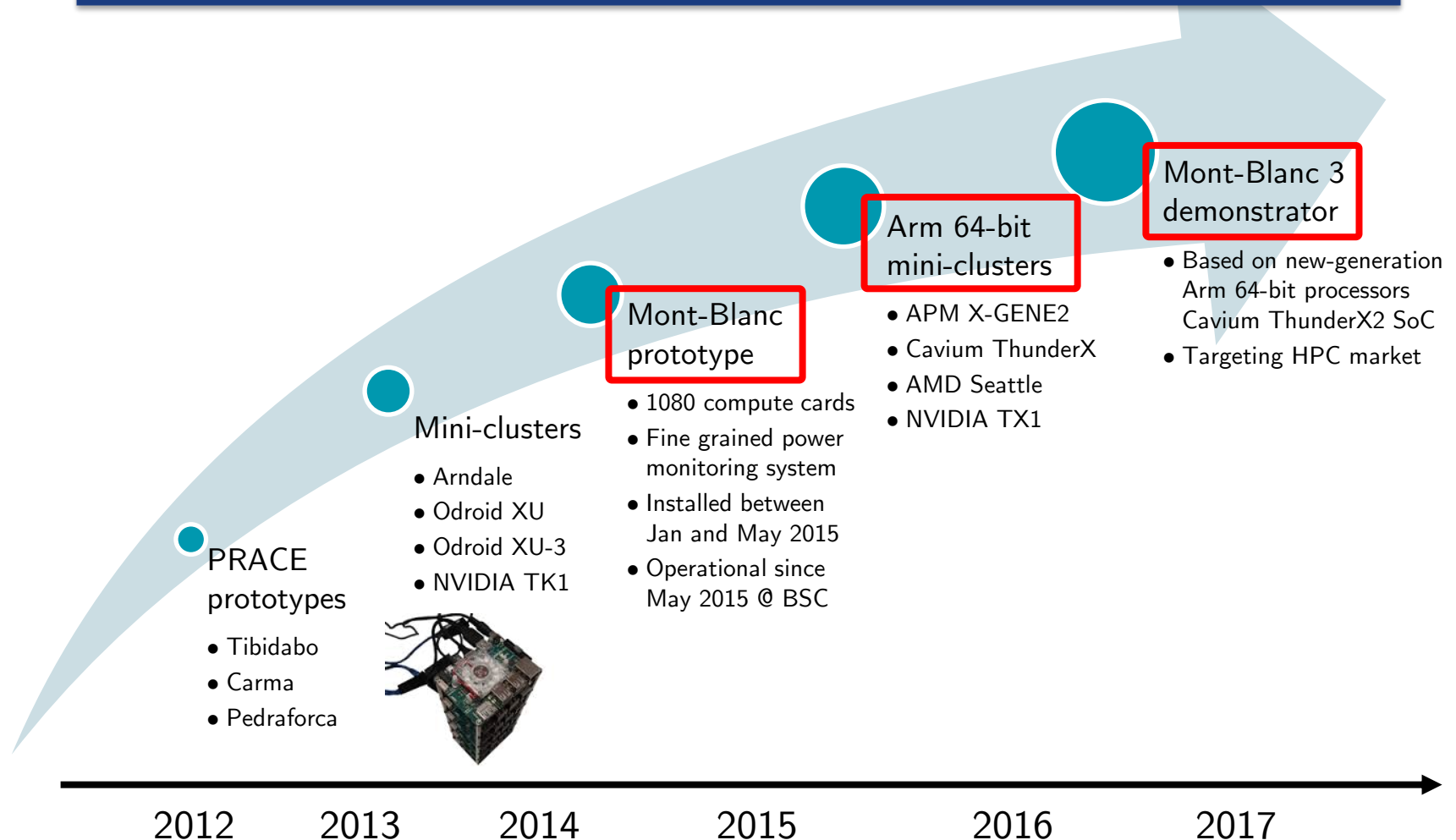
Other challenges

- Educational challenges
- Business challenges

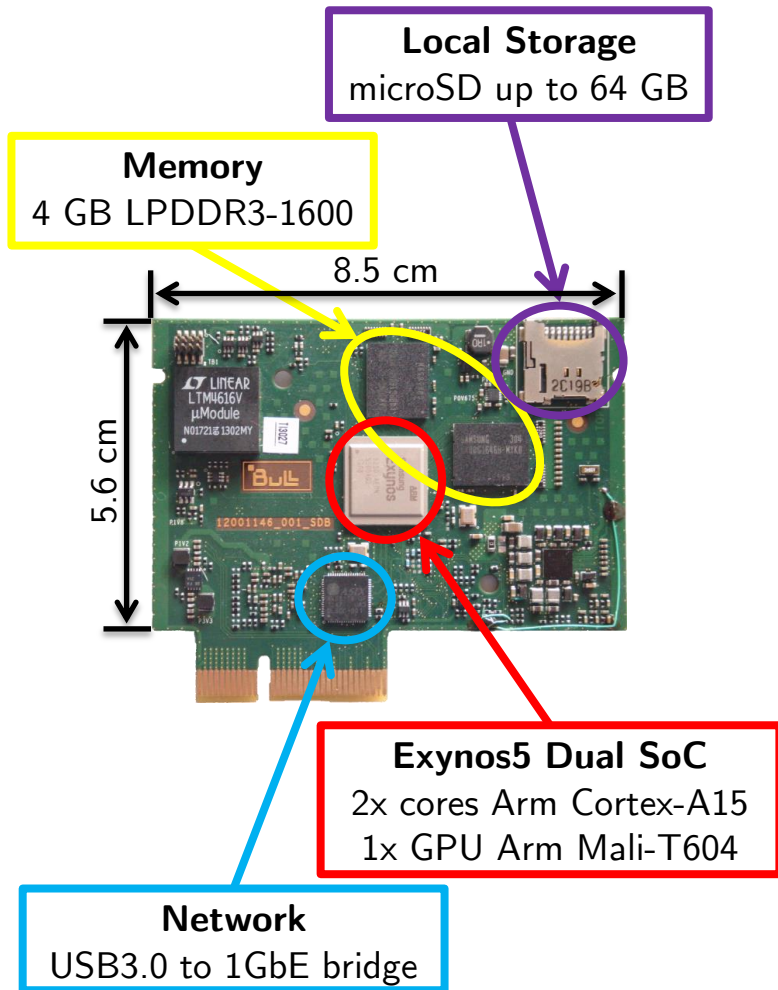


The Mont-Blanc prototype ecosystem

Prototypes are critical to accelerate software development
System software stack + applications



The first Mont-Blanc prototype



2 Racks	2160 CPUs
8 BullX chassis	1080 GPUs
72 Compute blades	4.3 TB of DRAM
1080 Compute cards	17.2 TB of Flash

Operational since May 2015 @ BSC

Mont-Blanc mini-clusters

→ Goal

- Tracking the evolution of Arm SoCs for their use in HPC

→ Method

- Exploring off-the-shelves technologies
- Liaising with SoC providers and system integrators



→ Implementation

- 16 nodes Odroid XU3
- 8 nodes NVIDIA TegraK1
- 15 nodes NVIDIA TegraX1
- 4 nodes AppliedMicro X-GENE2
- 5 nodes Cavium ThunderX
- 1 node AMD Seattle

6 Arm-based mini-clusters
up and running 24/7,
deployed at BSC and available
for the scientific community

Dibona: the Mont-Blanc 3 demonstrator



Mont-Blanc contributions

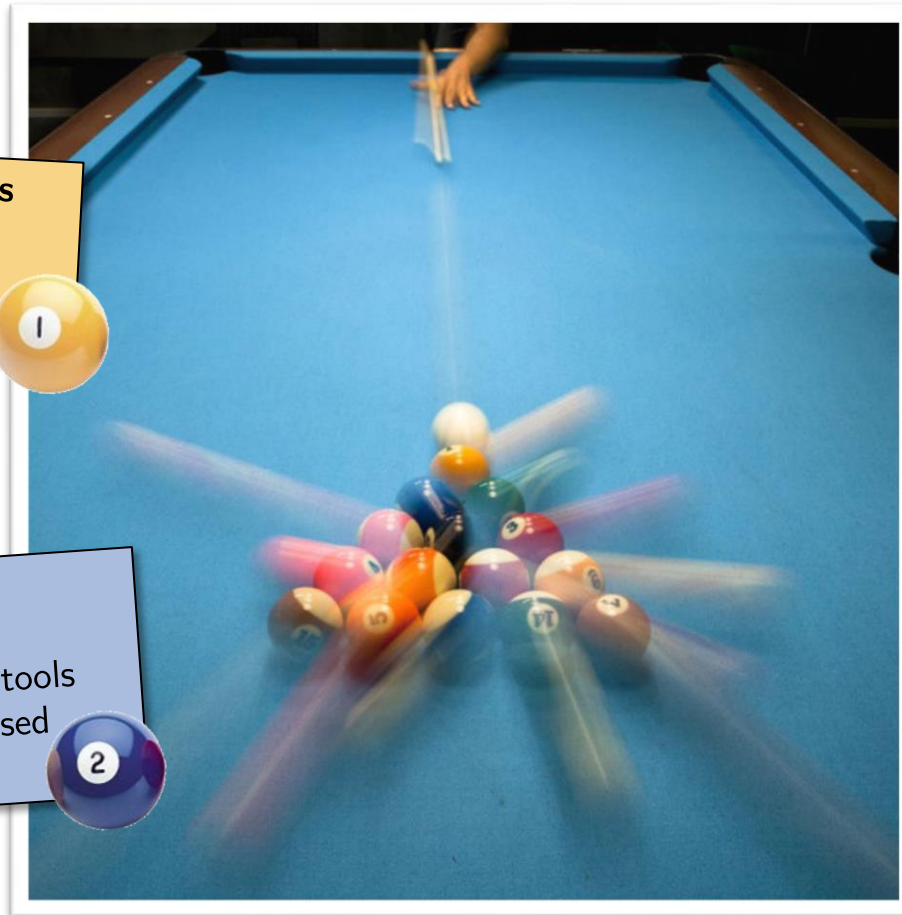
Arm-based prototypes

- Mobile technology
- Server technology
- System integration

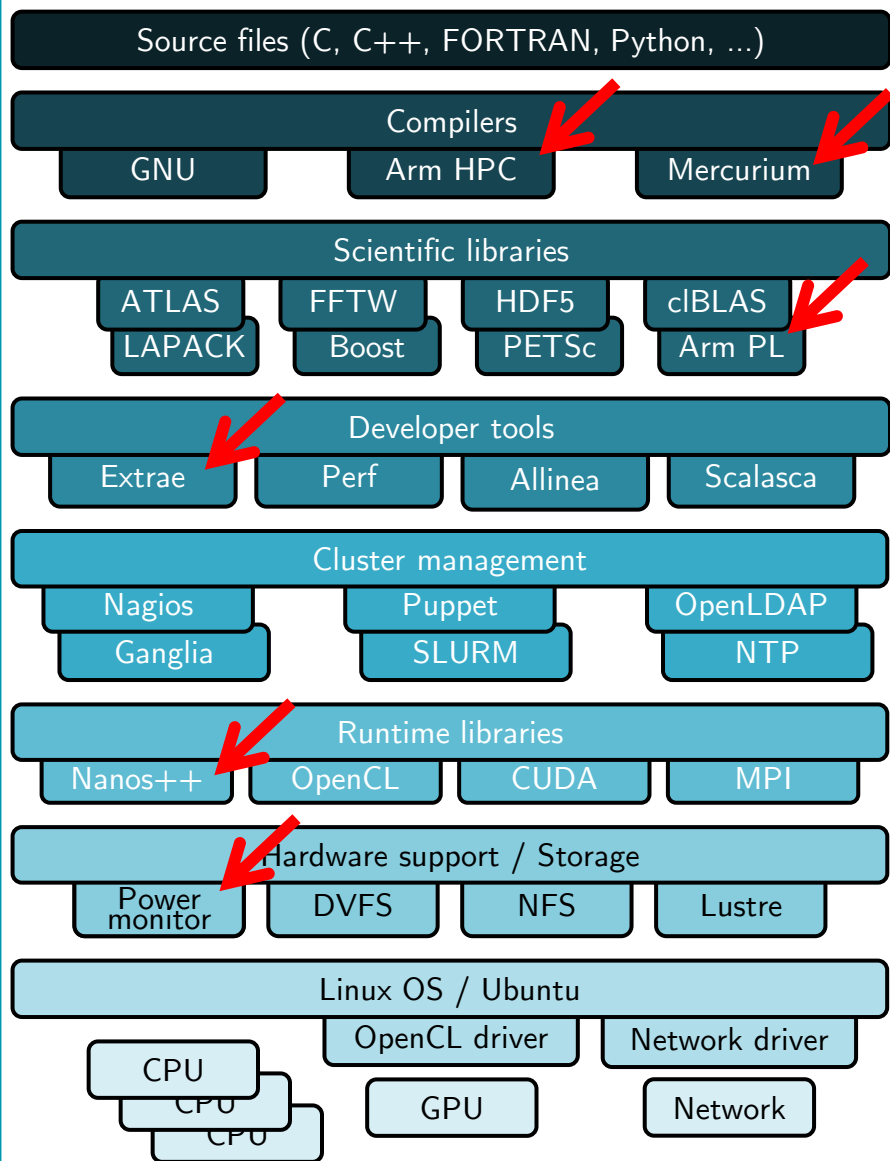


System software

- Programming model
- Performance analysis tools
- Evaluation of Arm-based ecosystem for HPC



System software stack for Arm



1

Developed since 2011!
Today in collaboration with all major OpenHPC partners

2

Tested on several Arm-based platform

3

Mostly based on open-source packages

4

Overlapping completely with "standard" HPC system software stack

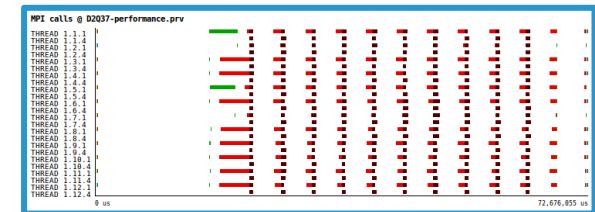
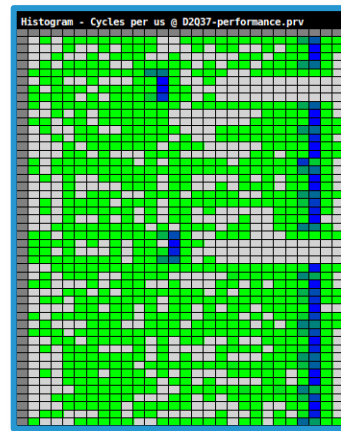
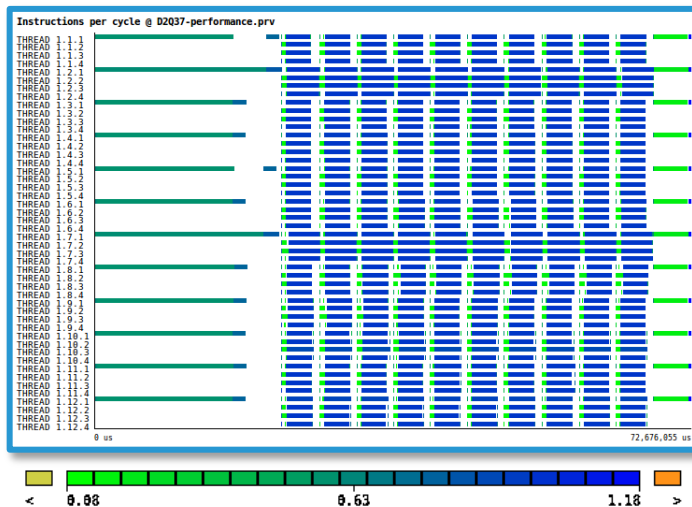
BSC #2: performance analysis tools

→ Extrae: binary instrumentation

- ./trace.sh you-binary → Run you application and generate a trace
- Traces are collection of timestamped events
- In the trace are collected several events specified in a xml config file
 - Beginners like me mostly get PAPI counters

→ Paraver: graphical trace visualizer

- Post-mortem analysis
- Allow analysis applying different semantics / filters / histograms



	Outside MPI	MPI_Barrier	MPI_Reduce	MPI_Comm_rank	MPI_Comm_size	MPI_Init	MPI_Finalize	MPI_Sendrecv
THREAD 1.1.1	73.26 %	13.38 %	5.51 %	0.00 %	0.00 %	0.01 %	0.16 %	7.48 %
THREAD 1.1.2	100 %	-	-	-	-	-	-	-
THREAD 1.1.3	100 %	-	-	-	-	-	-	-
THREAD 1.1.4	92.45 %	-	-	-	-	-	-	7.55 %
THREAD 1.2.1	91.54 %	0.81 %	0.00 %	0.00 %	0.00 %	0.03 %	0.01 %	7.61 %
THREAD 1.2.2	100 %	-	-	-	-	-	-	-
THREAD 1.2.3	100 %	-	-	-	-	-	-	-
THREAD 1.2.4	92.32 %	-	-	-	-	-	-	7.68 %

A image trace is worth a thousand words

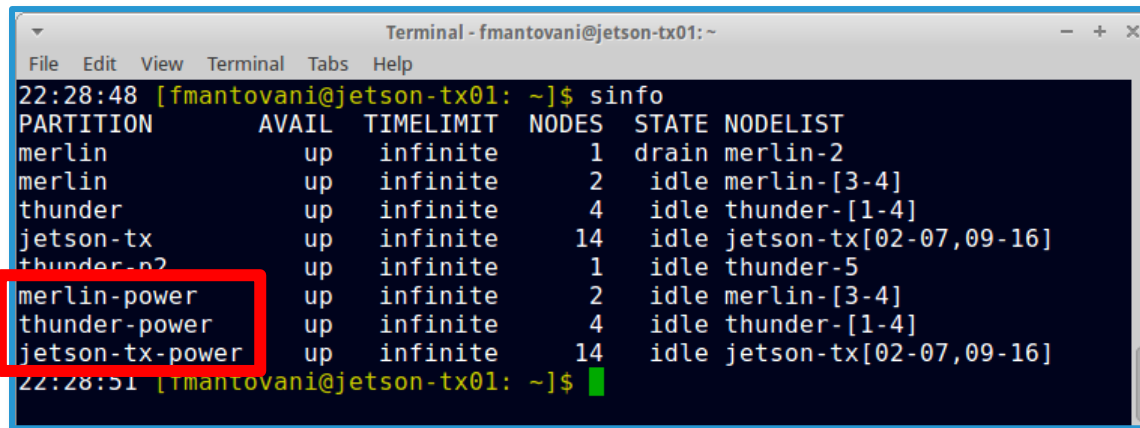
BSC #3: Power traces on mini-clusters

→ Goal

- To make power monitoring easy for the user

→ Implementation

- SLURM integration
- Launching jobs to *-power queue
- Get power traces in format CSV and Paraver



```
Terminal - fmantovani@jetson-tx01: ~  
File Edit View Terminal Tabs Help  
22:28:48 [fmantovani@jetson-tx01: ~]$ sinfo  
PARTITION      AVAIL  TIMELIMIT  NODES  STATE NODELIST  
merlin          up    infinite    1     drain merlin-2  
merlin          up    infinite    2     idle  merlin-[3-4]  
thunder         up    infinite    4     idle  thunder-[1-4]  
jetson-tx       up    infinite   14     idle  jetson-tx[02-07,09-16]  
thunder-p2      up    infinite    1     idle  thunder-5  
merlin-power    up    infinite    2     idle  merlin-[3-4]  
thunder-power  up    infinite    4     idle  thunder-[1-4]  
jetson-tx-power up    infinite   14     idle  jetson-tx[02-07,09-16]  
22:28:51 [fmantovani@jetson-tx01: ~]$
```

→ Observation: power measurements are highly heterogeneous

- Coming from different devices (in-band, out-of-band, ...)
- With different sampling rates
- With synchronization issues

Mont-Blanc and the Arm HPC ecosystem

→ Compilers

- GCC
- Arm HPC Compiler (based on LLVM)

In classical HPC systems
e.g. Intel Compiler



→ Math libraries

- OpenBLAS, ATLAS, FFTw, ...
- Arm Performance Libraries

In classical HPC systems
e.g. Intel Math Kernel Library



How mature is the Arm HPC ecosystem?

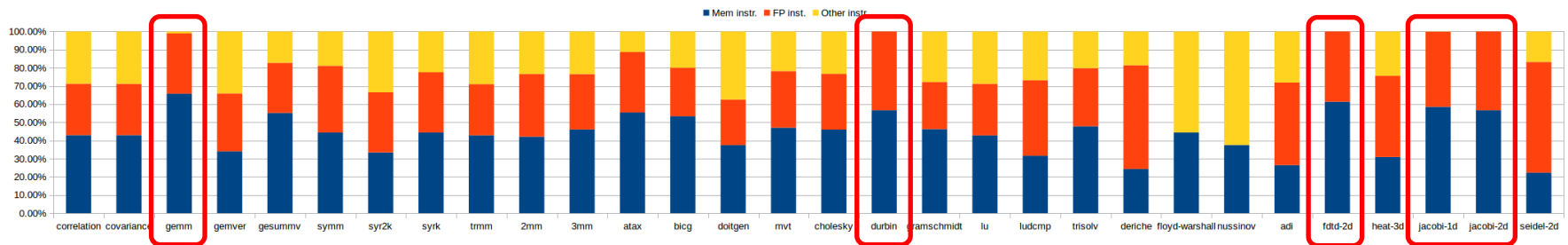
Evaluation of the Arm HPC Compiler

→ Polybench benchmark suite

- Including 30 benchmarks by Ohio State University

→ Method

- Compile with Arm HPC Compiler
- Run and look at the instruction mix in Cortex-A57 (reference Armv8)
- Focus on benchmarks where FP and Mem instruction are dominant (>90%)





Which are the differences with standard HPC systems?
Which compilers generate less instructions?

Total number of instructions

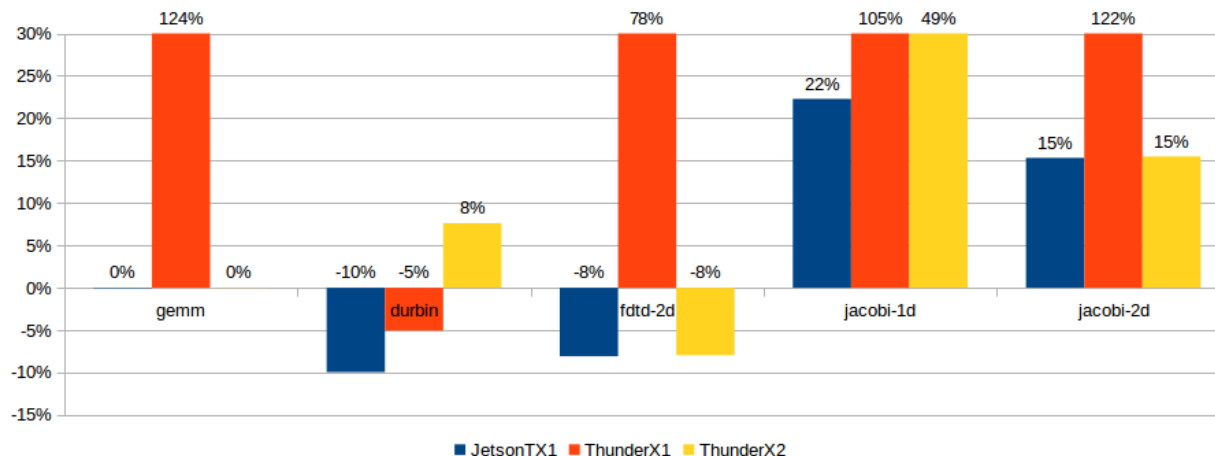
→ Expectation

- RISC (Arm) more instructions than CISC (Intel)

→ Method

- Considering the subset of Polybench 
- Compile them with GCC v7.1.0 
- Run them on 3 Arm SoCs and compare with Intel Skylake (MareNostrum4)
 - Arm #1: NVIDIA JetsonTX1
 - Arm #2: Cavium ThunderX1
 - Arm #3: Cavium ThunderX2

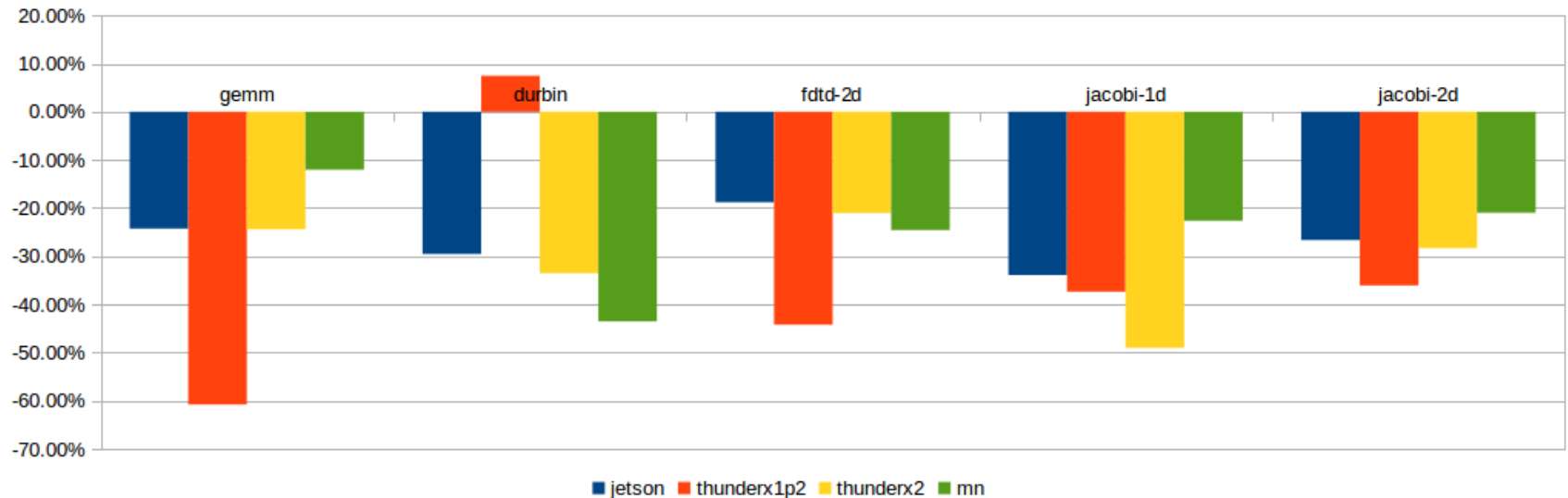
What about proprietary compilers?



GCC vs proprietary compilers

→ Method

- Considering the subset of Polybench
- Compile them with GCC v7.1.0, Arm HPC Compiler 1.4 and Intel ICC 17.0.4
- Run them on 3 Arm SoCs and on Intel Skylake (MareNostrum4)



Is this always true?

Arm HPC Compiler + OpenMP

→ Comparison of GCC 7.1.0 vs Arm HPC Compiler 1.3

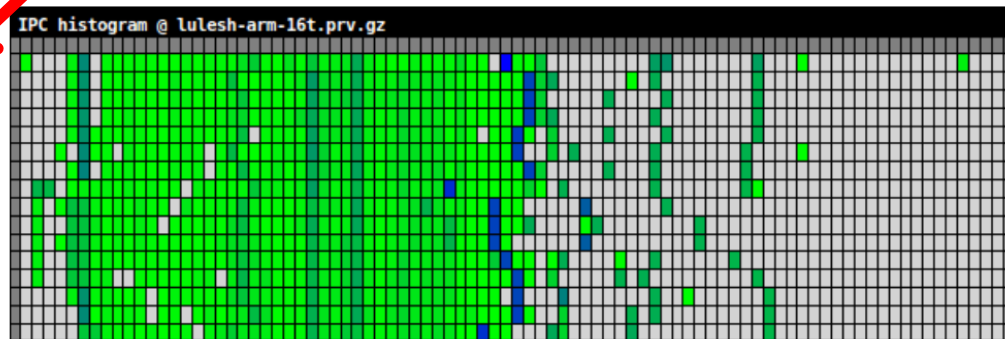
- Arm HCP Compiler 1.4 has been released on August 17 ☹

→ Evaluation on Polybench, LULESH, CoMD and QuantumESPRESSO

- Preliminary results show huge difference in number of instructions

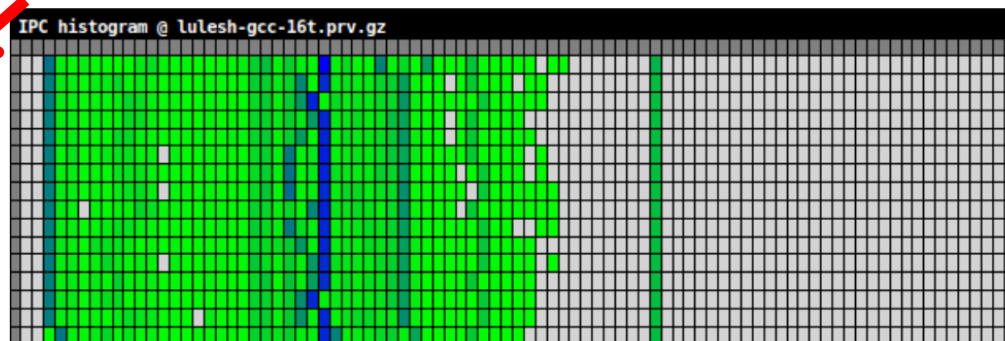
Overall figures

	Value
Total Instructions	4.36×10^9
Average Per Thread	0.27×10^9
Avg/Max	0.93
Total Execution Time	51.35 s
Time Per Iteration	495.29 ms



Overall figures

	Value
Total Instructions	2.99×10^9
Average Per Thread	0.19×10^9
Avg/Max	0.93
Total Execution Time	48.27 s
Time Per Iteration	393.04 ms



More details in the poster accepted at SC'17

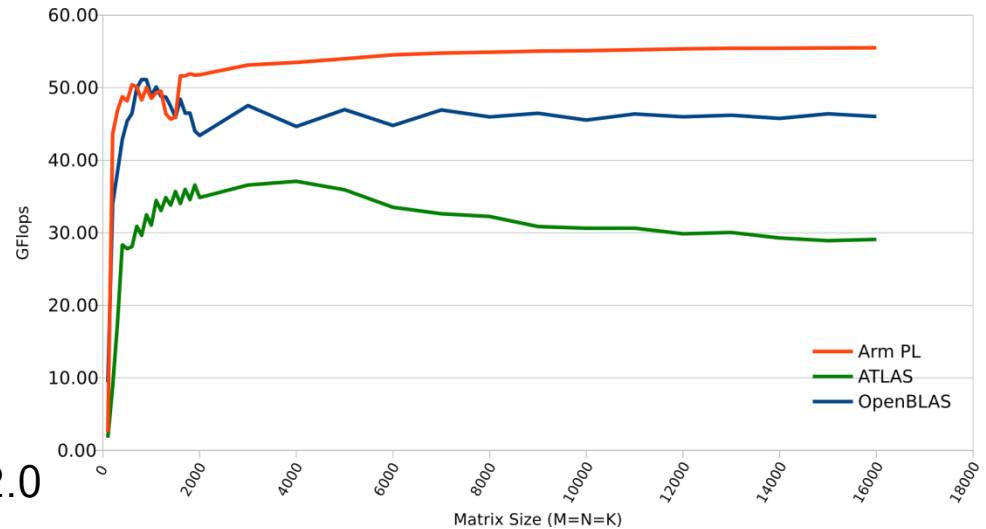
Evaluation of the Arm Performance Libraries

→ DGEMM micro-kernel

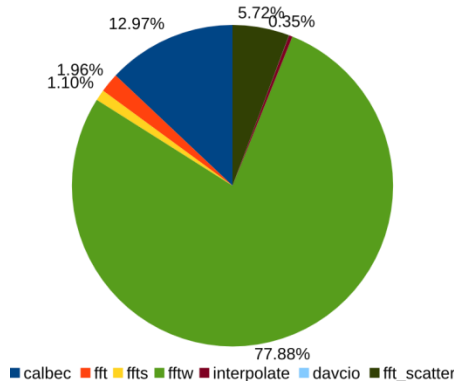
- AMD Seattle
- It reaches 87% of the peak with matrix 16000x16000

→ QuantumESPRESSO

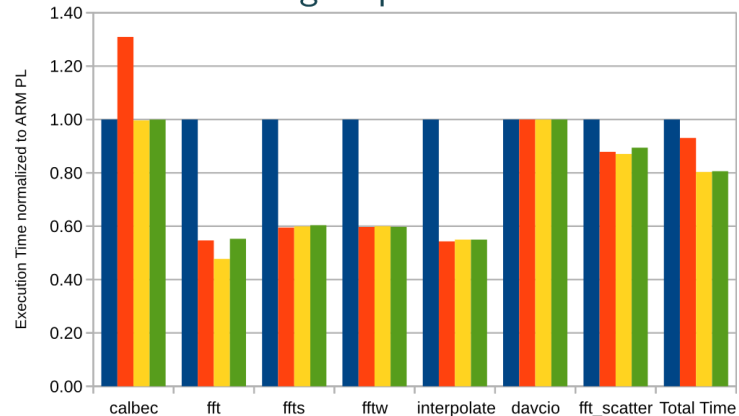
- Arm Performance Libraries 2.2.0
- ATLAS 3.11.39 + FFTW 3.3.6
- OpenBLAS 0.2.20 + FFTW 3.3.6
- Arm Performance Libraries 2.2.0 + FFTW 3.3.6



Execution time breakdown for the large input size



Large input size



Take away message

Yes, ISAs are different, but when adding levels of complexity...

Compilers (GCC, proprietary, ...)

Parallelization runtimes and libraries (OpenMP, MPI)

Math libraries (BLAS, ATLAS, FFTw, ...)

... you end up not appreciating the differences between ISAs



Suggested lecture:

E. Blem, J. Menon, and K. Sankaralingam,
“Power struggles: Revisiting the RISC vs. CISC debate on
contemporary ARM and x86 architectures,”
in 2013 IEEE 19th International Symposium on High Performance
Computer Architecture (HPCA), 2013, pp. 1–12.

Mont-Blanc contributions

Arm-based prototypes

- Mobile technology
- Server technology
- System integration



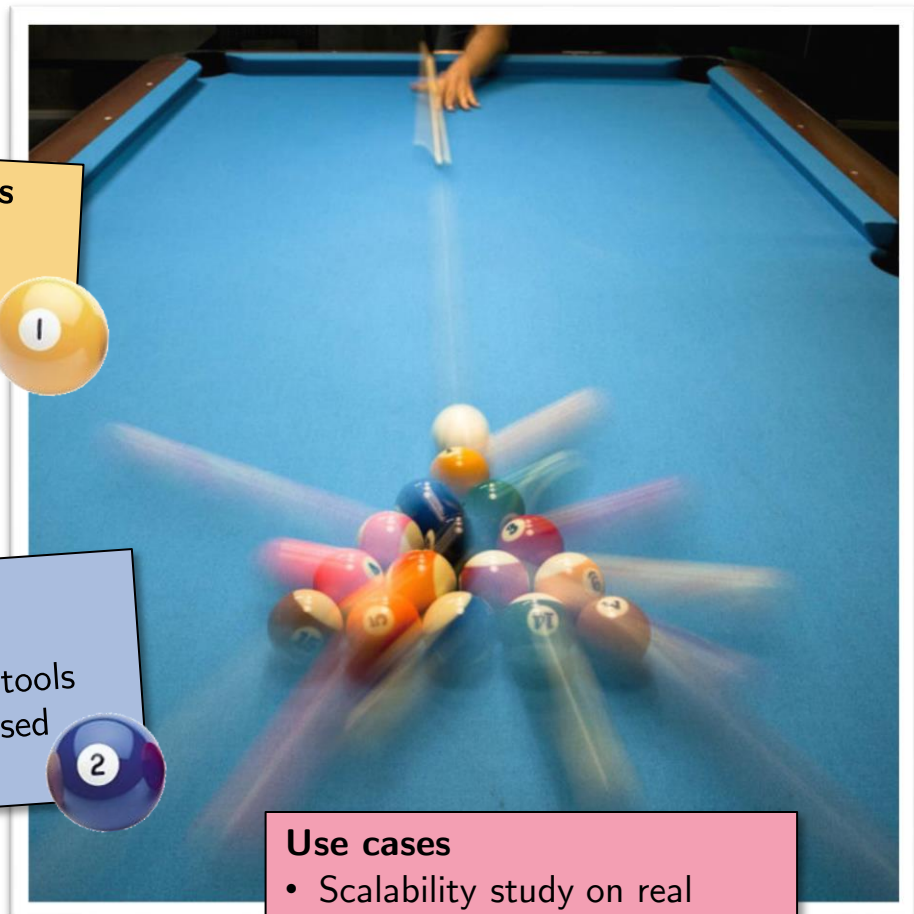
System software

- Programming model
- Performance analysis tools
- Evaluation of Arm-based ecosystem for HPC

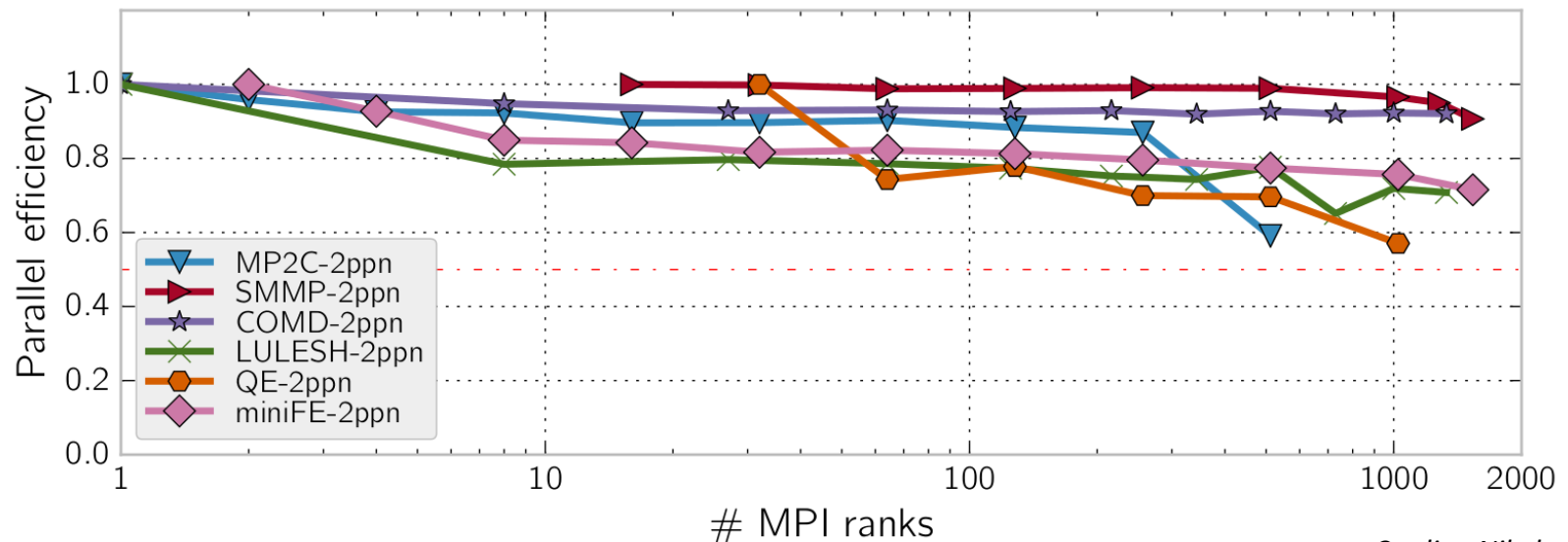
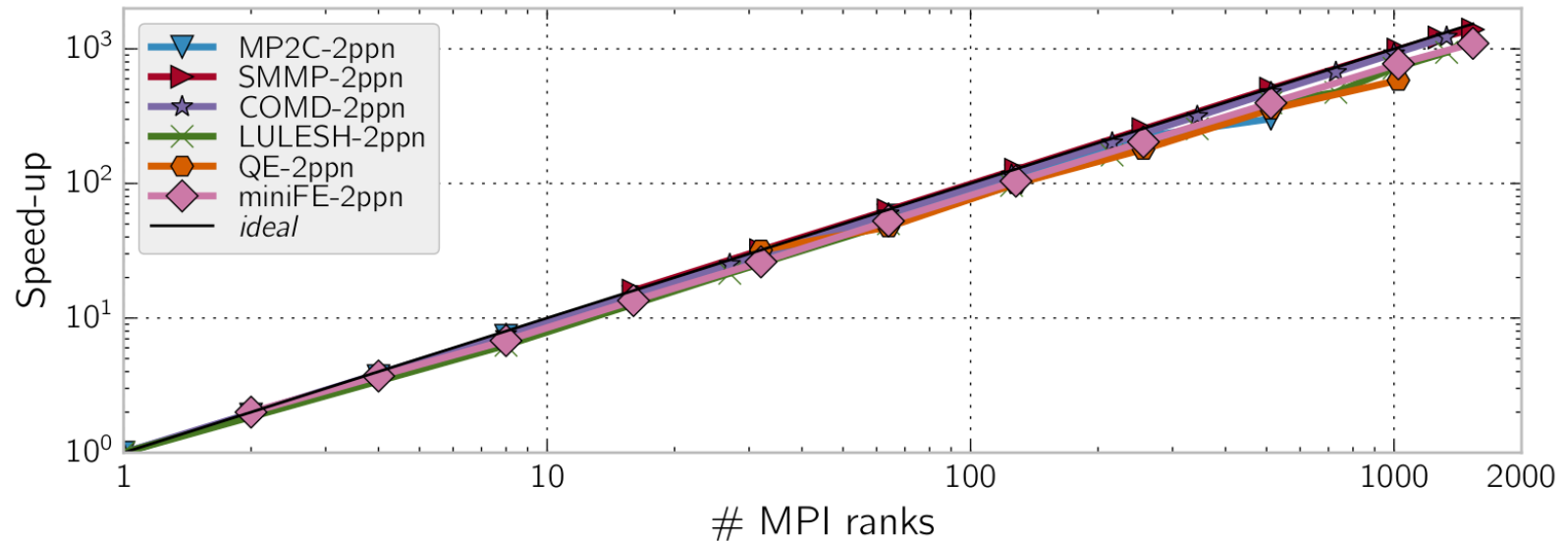


Use cases

- Scalability study on real Arm-based platforms
- Correlation of performance and power measurements
- Runtime features for future HPC systems

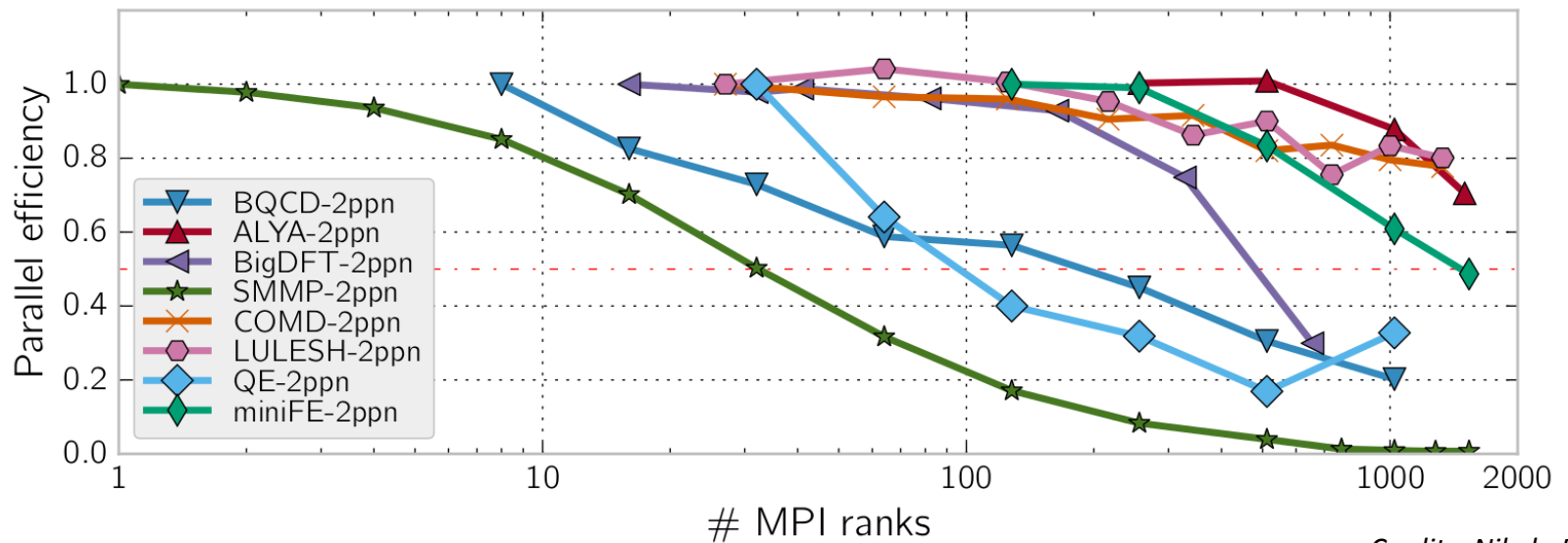
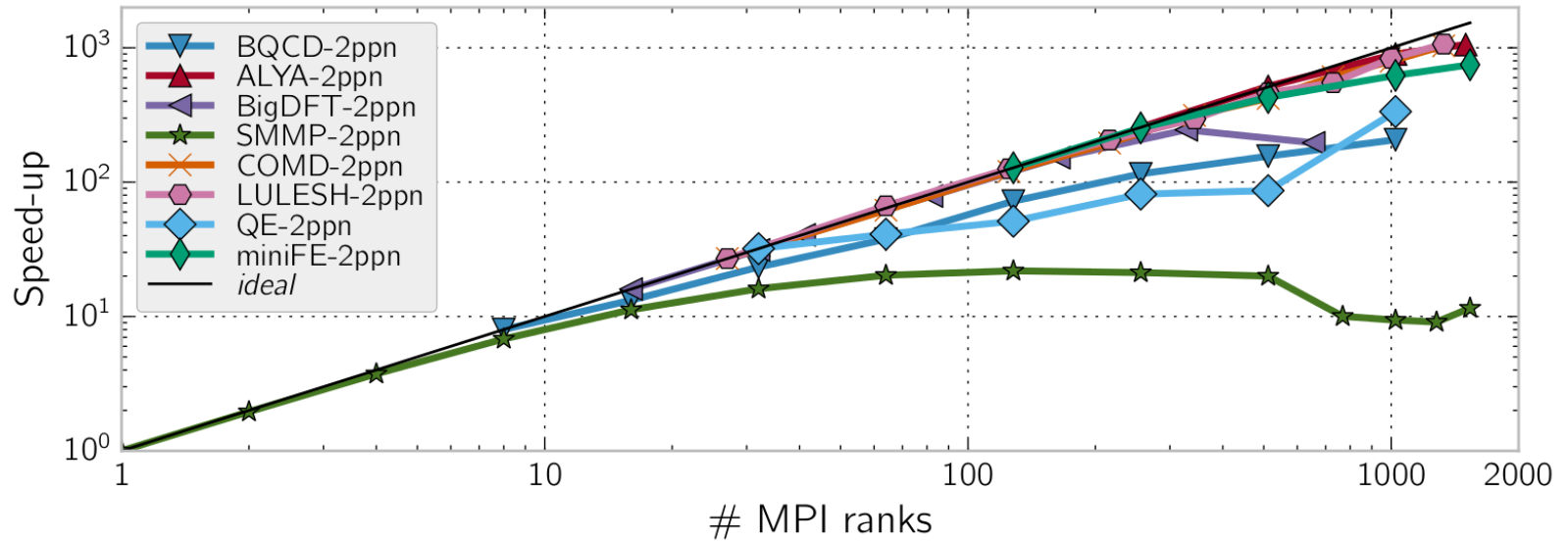


Weak scaling study on mobile technology



Credits: Nikola Rajovic

Strong scaling study on mobile technology



Credits: Nikola Rajovic

→ Applications

- Benchmarks
- Mini-apps
- Production / Industrial codes

→ Tracing applications with the objective of...

- Test current solutions and provide feedbacks to technology providers
 - Evaluation of Arm HPC Compiler and Arm Performance Libraries
 - Measure power consumption and correlate it with performance
- Understanding code limitations and helping the developers in restructuring it applying OmpSs/OpenMP4.0 and analyze the effect
 - Benefit of taskification
 - Exploring new techniques, e.g. Dynamic Load Balancing
- Performing extrapolation studies using next generation machine parameters
 - MULTIscale Simulation Architecture

Lattice Boltzmann D2Q37

→ Fluid dynamic code MPI+OpenMP for simulation of e.g. mixing layer evolution of fluids at different temperature/density

→ Simple structure

- Serial initialization + closing
- Propagate (memory bound)
- Collide (compute bound)

Propagate

Collide

IPC @ D2Q37.prv

THREAD 1.1.1
THREAD 1.1.2
THREAD 1.1.3
THREAD 1.1.4
THREAD 1.1.5
THREAD 1.1.6
THREAD 1.1.7
THREAD 1.1.8
THREAD 1.1.9
THREAD 1.1.10
THREAD 1.1.11
THREAD 1.1.12
THREAD 1.1.13
THREAD 1.1.14
THREAD 1.1.15

Initialization

Closing

THREAD 1.1.23
THREAD 1.1.24
THREAD 1.1.25
THREAD 1.1.26
THREAD 1.1.27
THREAD 1.1.28
THREAD 1.1.29
THREAD 1.1.30
THREAD 1.1.31
THREAD 1.1.32
THREAD 1.1.33
THREAD 1.1.34
THREAD 1.1.35
THREAD 1.1.36
THREAD 1.1.37
THREAD 1.1.38
THREAD 1.1.39
THREAD 1.1.40
THREAD 1.1.41
THREAD 1.1.42
THREAD 1.1.43
THREAD 1.1.44
THREAD 1.1.45
THREAD 1.1.46
THREAD 1.1.47
THREAD 1.1.48

442,664,351 us

536,894,325 us

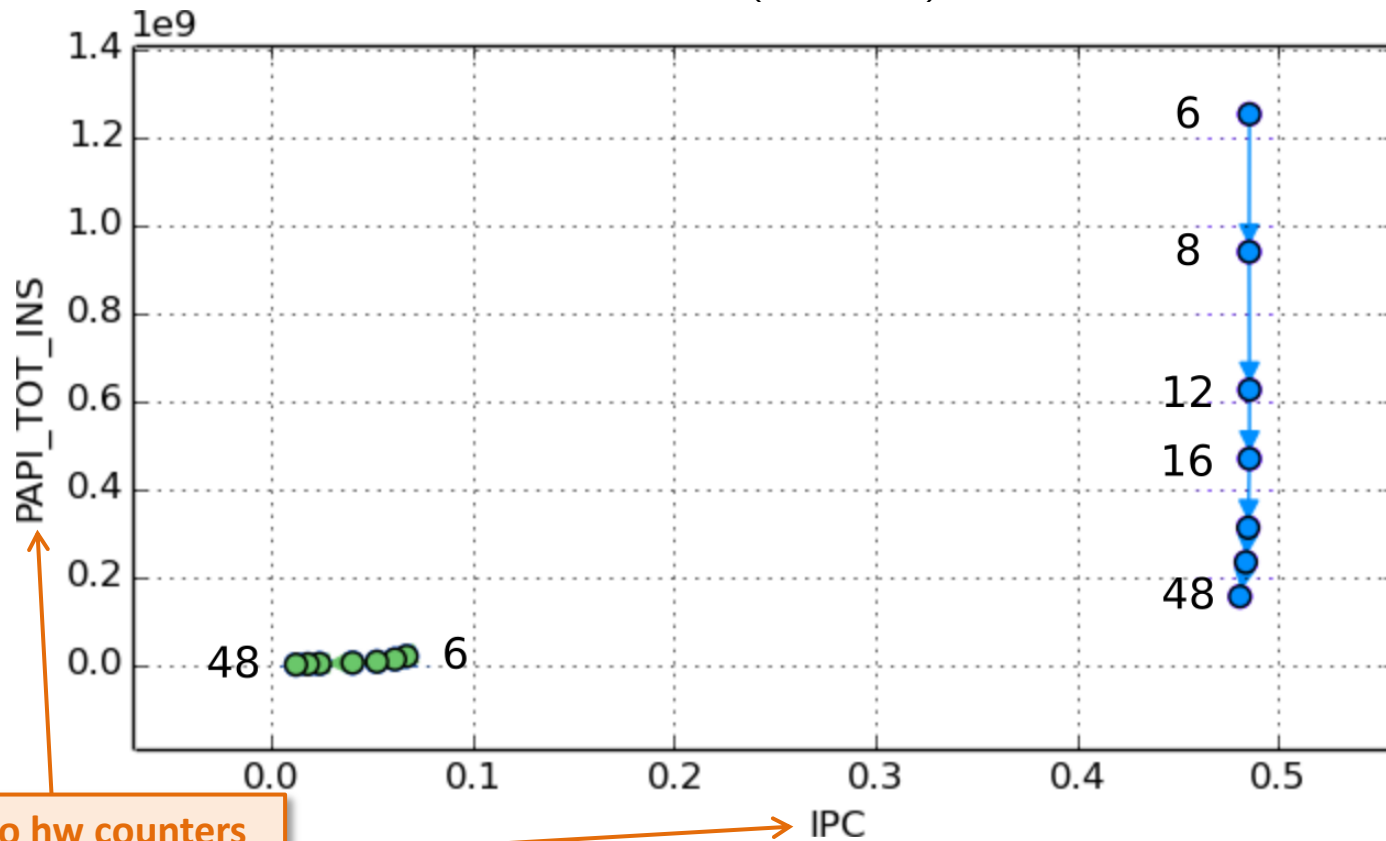


Credits: E. Calore

D2Q37 Clustering analysis (on ThunderX)

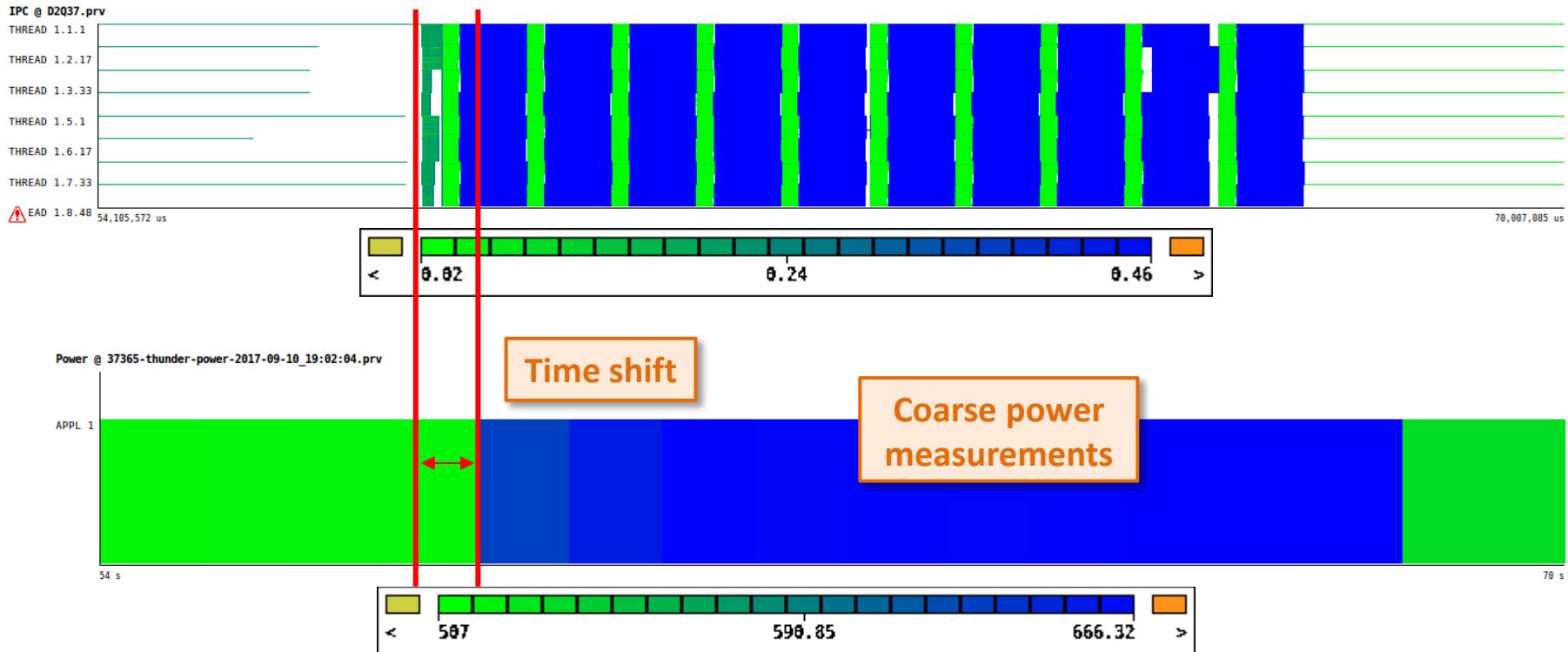
→ Different runs have been performed over the same lattice size with a varying number of threads: 6, 8, 12, 16, 24, 32, 48.

- Collide function is scaling almost perfectly up to 48 threads
- For the Propagate, increasing the number of threads makes threads competing for the same resource (memory)



Access to hw counters allows this kind of study

D2Q37 Correlating performance, power, energy



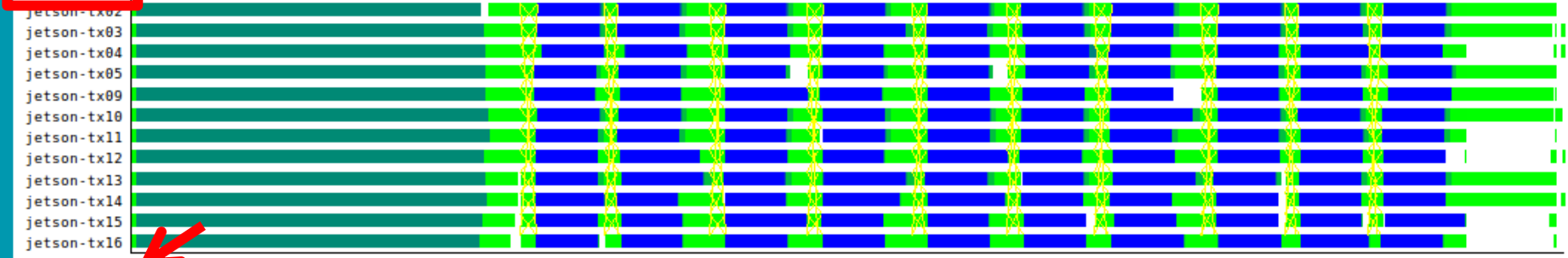
➔ Even if with limitations, the user can access power figures

- Timeline of “instantaneous” power consumption
- Energy to solution in one click

D2Q37 Correlating performance, power, energy

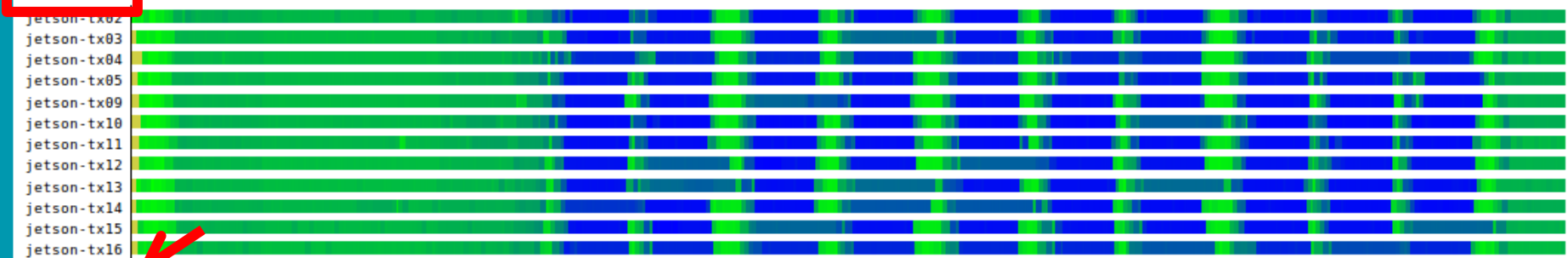
GFLOP (vec)

D2Q37.prv



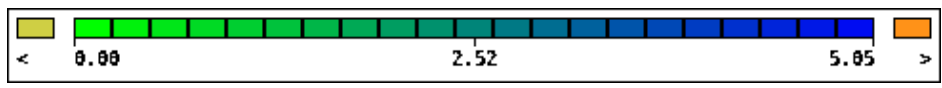
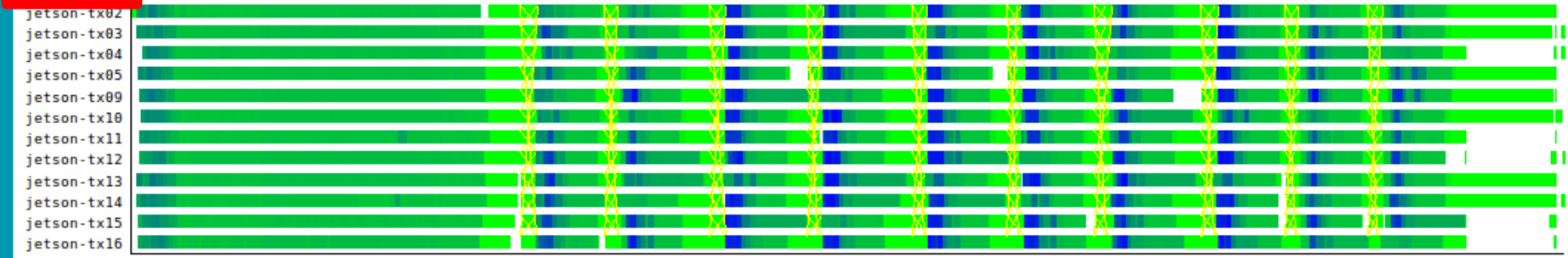
Power @ 3734

-jetson-tx-power-2017-09-08_17:07:42.prv

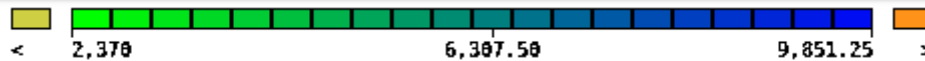
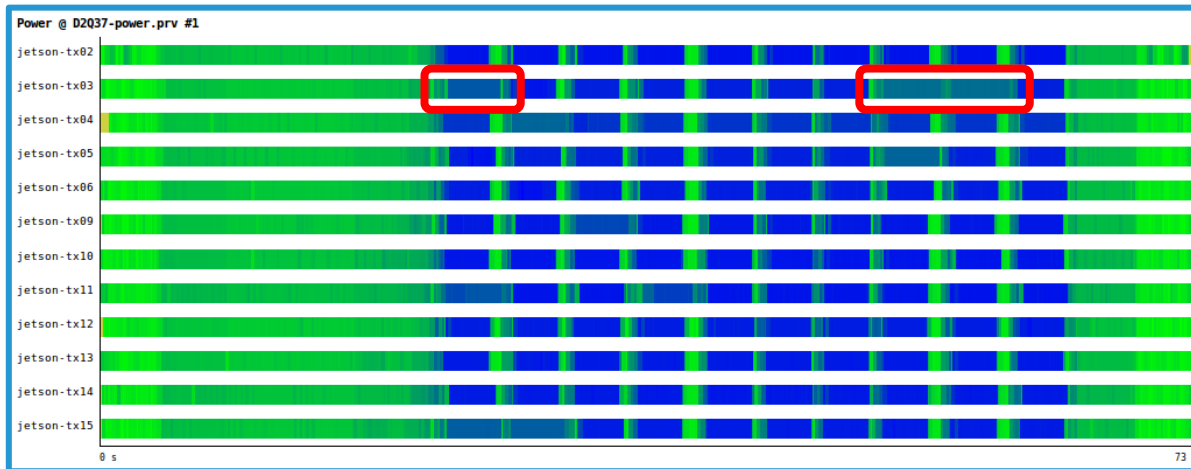
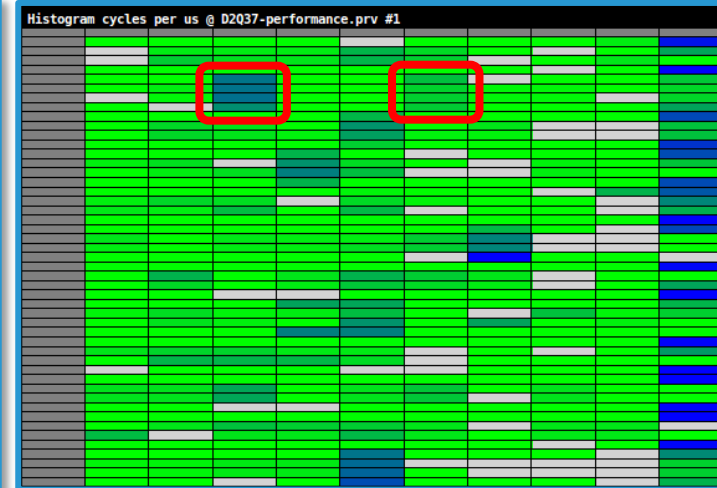
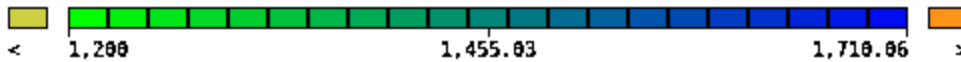
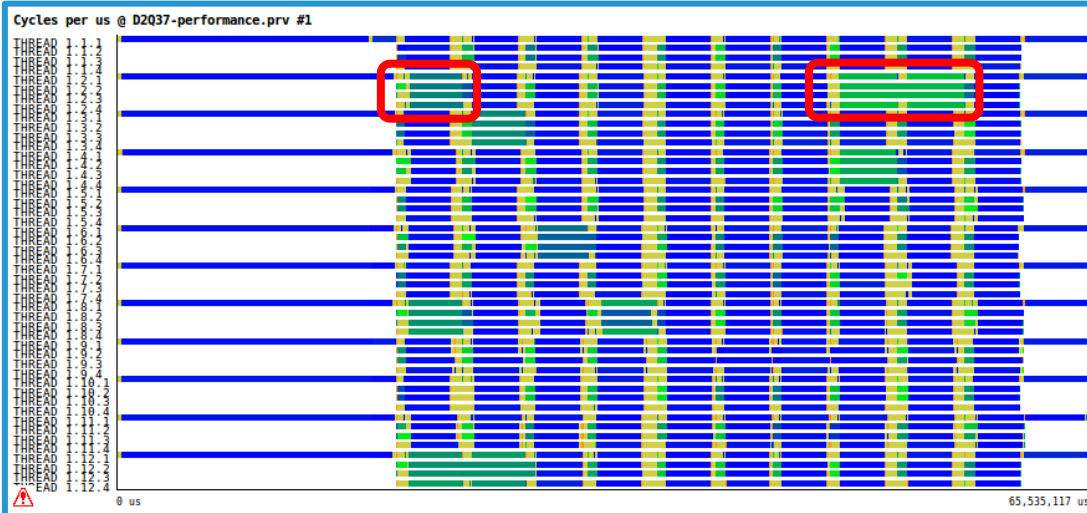


GFLOP/W @ 37

49-jetson-tx-power-2017-09-08_17:07:42.prv



Frequency analysis and energy to solution



Energy to solution @ D2037-powe + x

[2,370.00..10,638.75]

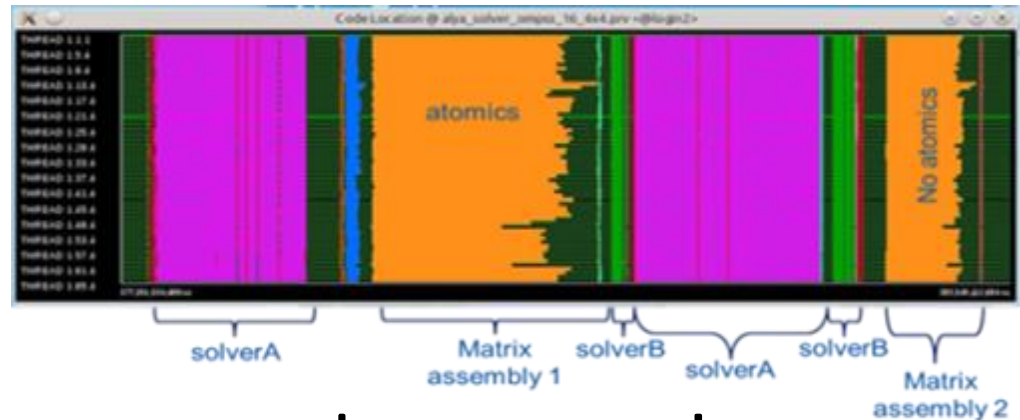
jetson-tx02	469,427.16
jetson-tx03	439,638.69
jetson-tx04	430,995.02
jetson-tx05	453,455.81
jetson-tx06	458,372.55
jetson-tx09	457,328.90
jetson-tx10	469,052.97
jetson-tx11	469,194.46
jetson-tx12	456,111.53
jetson-tx13	459,625.07
jetson-tx14	469,675.26
jetson-tx15	450,010.75
Total	5,482,888.18
Average	456,907.35
Maximum	469,675.26
Minimum	430,995.02
StDev	11,753.48
Avg/Max	0.97



Alya: BSC code for multi-physics problems

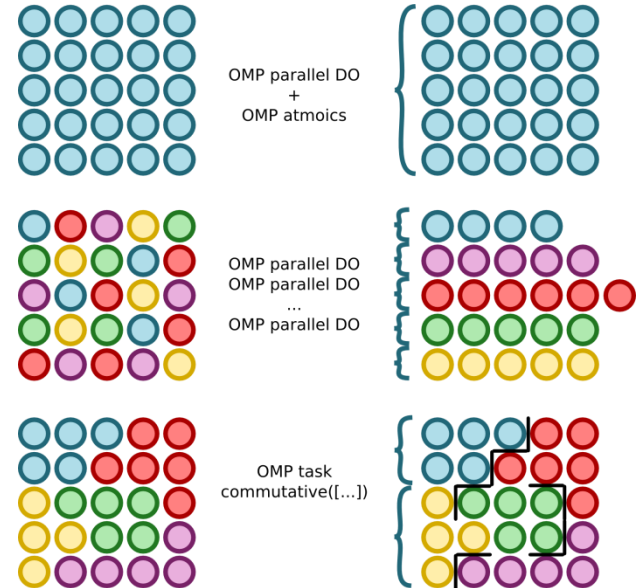
Parallelization of finite elements code

→ Analysis with Paraver:



→ Reductions with indirect accesses on large arrays using

- No coloring
Use of atomics operations harms performance
- Coloring
Use of coloring harms locality
- Commutative Multidependences
(OmpSs feature to be hopefully included in OpenMP)



Credits: M. Garcia, J. Labarta

Alya: taskification and dynamic load balancing

→ Towards throughput computing

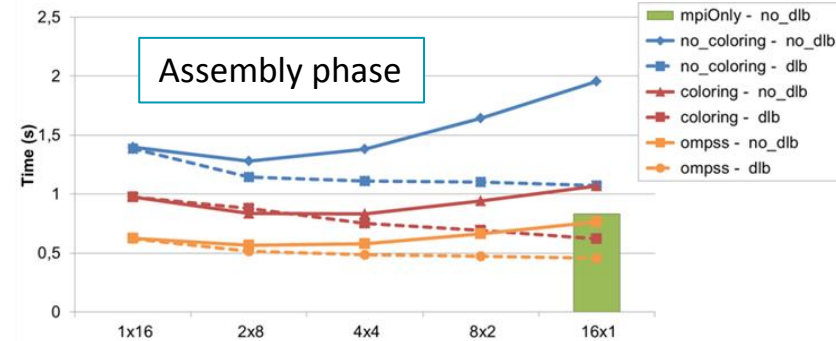
- Tasks + DLB → dotted lines

→ DLB helps in all cases

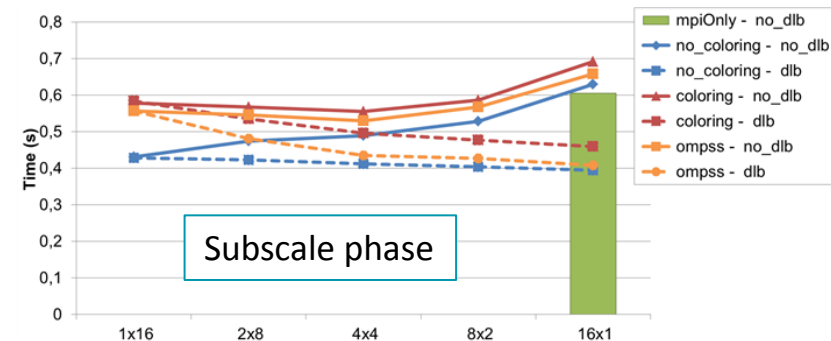
- Even more in the bad ones

→ Side effects

- Hybrid MPI+OmpSs Nx1 can perform better than pure MPI!
- Nx1 + DLB: hope for lazy programmers



16 nodes x P processes/node x T threads/process



Mont-Blanc contributions

Arm-based prototypes

- Mobile technology
- Server technology
- System integration



System software

- Programming model
- Performance analysis tools
- Evaluation of Arm-based ecosystem for HPC



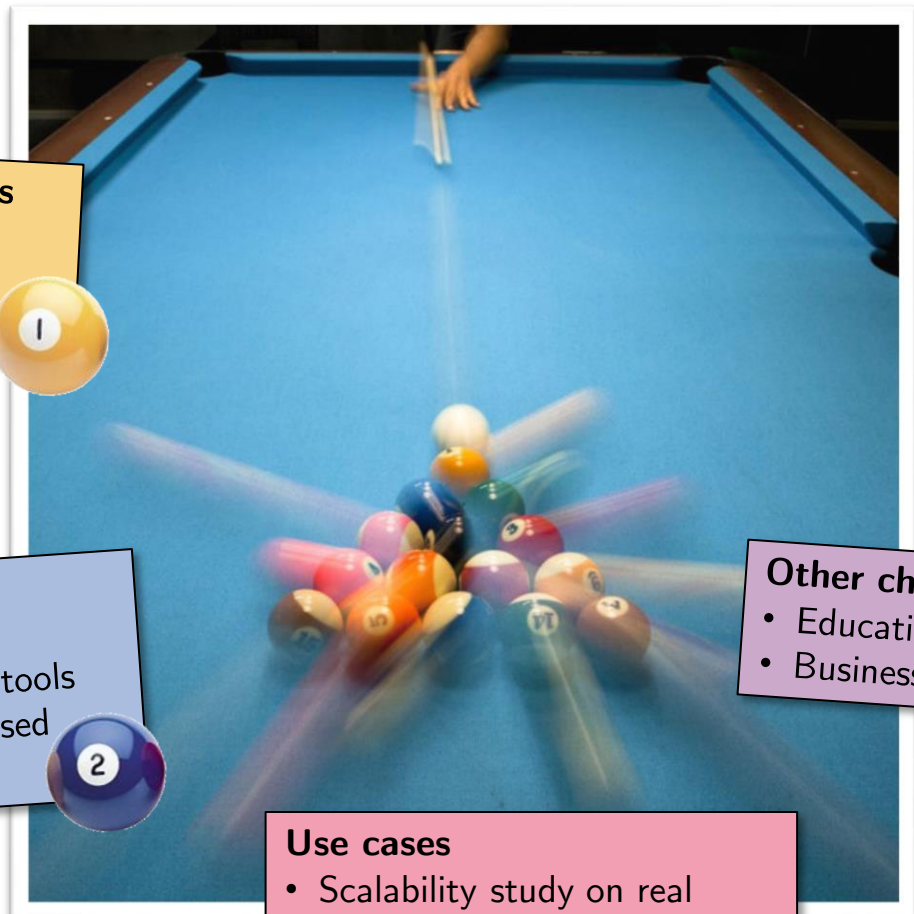
Use cases

- Scalability study on real Arm-based platforms
- Correlation of performance and power measurements
- Runtime features for future HPC systems



Other challenges

- Educational challenges
- Business challenges



Educational Challenge: Student Cluster Competition

→ Rules

- 12 teams of 6 undergraduate students
- 1 cluster operating within 3 kW power budget
- 3 HPC applications + 2 benchmarks

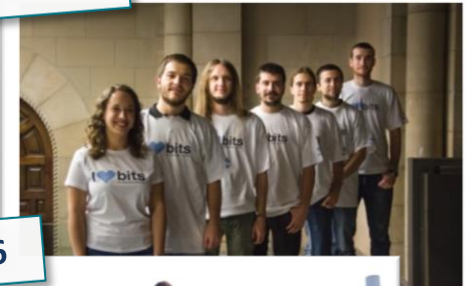
→ One team from University Politecnica of Catalunya (UPC-Spain)

- Participating with Mont-Blanc technology

→ 3 awards to win

- Best HPL
- 1st, 2nd, 3rd overall places
- Fan favorite

Team 2015



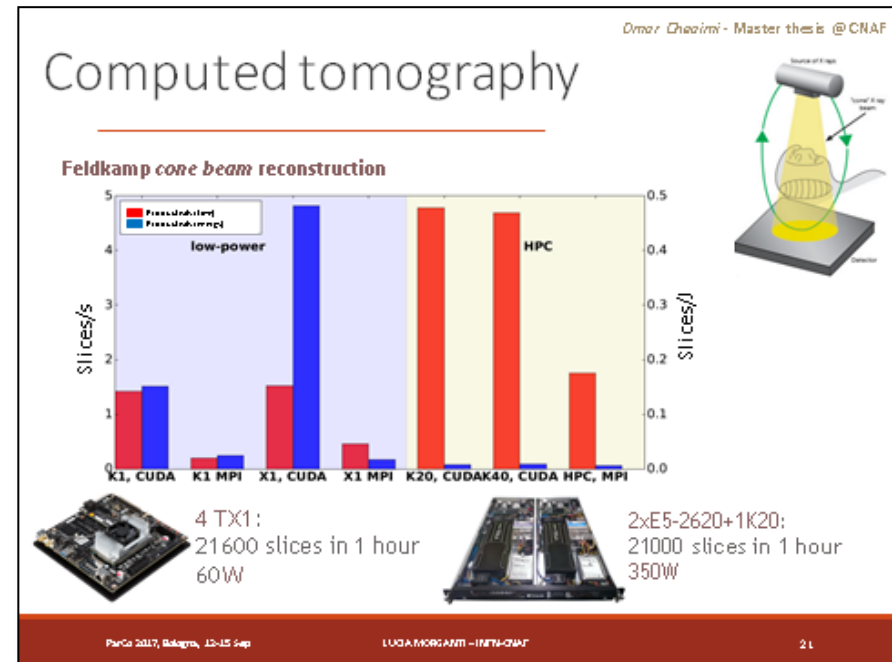
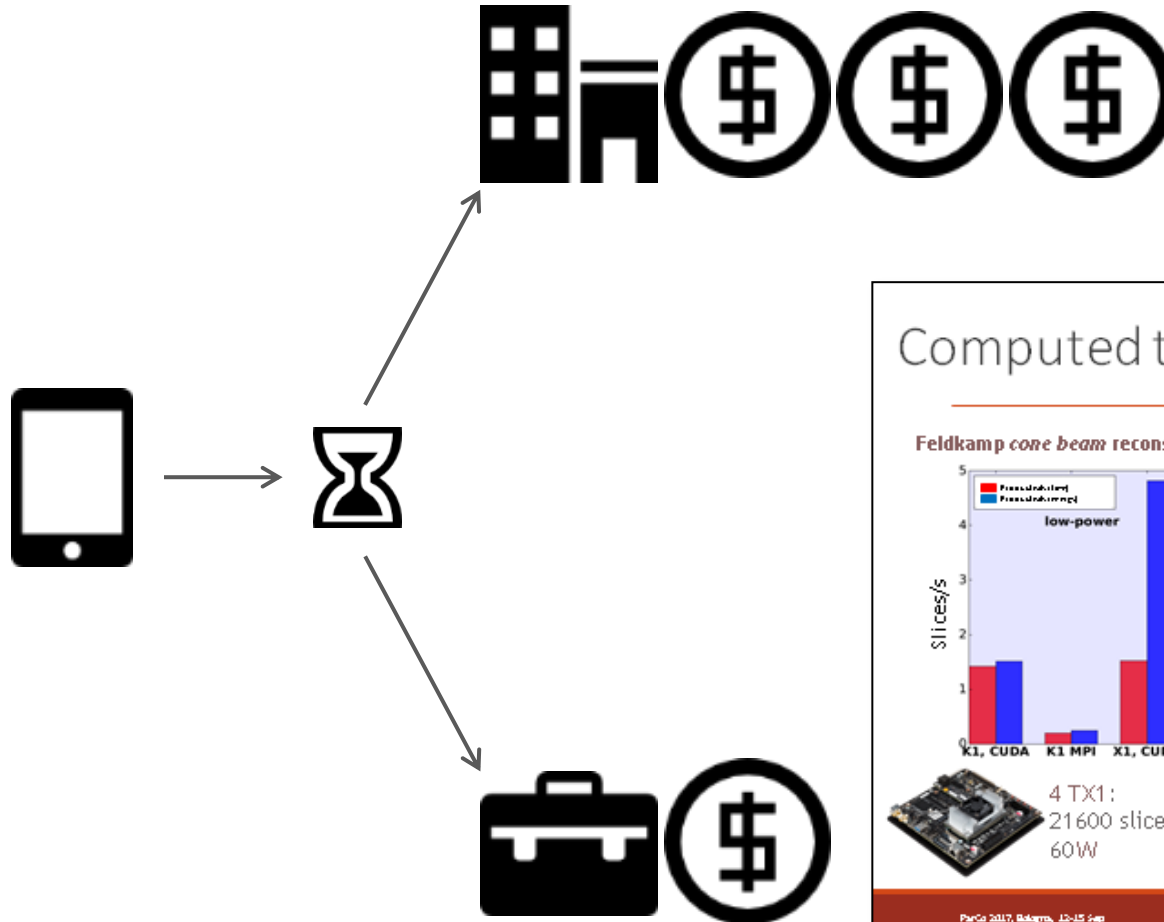
Team 2016



Team 2017



Business challenge: high end vs cost efficiency



Credits: E. Morganti, D. Cesini
<http://www.cosa-project.it/>

References

- i. **Evaluation of the first Mont-Blanc prototype**
N. Rajovic et al., “The Mont-Blanc Prototype: An Alternative Approach for HPC Systems,” in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Piscataway, NJ, USA, 2016, p. 38:1–38:12.
- ii. **Memory reliability study for the first Mont-Blanc prototype**
L. Bautista-Gomez et al., “Unprotected Computing: A Large-scale Study of DRAM Raw Error Rate on a Supercomputer,” in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Piscataway, NJ, USA, 2016, p. 55:1–55:11.
- iii. **PAPI support on Cavium and advanced performance analysis with Paraver**
D. Ruiz, E. Calore, and F. Mantovani, “Enabling PAPI support for advanced performance analysis on ThunderX SoC,” tech. report, <http://upcommons.upc.edu/handle/2117/107063>
- iv. **Performance analysis with Paraver**
D6.1 Report on profiling and benchmarking of the initial set of applications on ARM-based HPC systems
<http://bit.ly/mb3-d61>
- v. **Dynamic Load Balancing techniques and other programming model improvements**
D6.5 Initial report on automatic region of interest extraction and porting to OpenMP4.0-OmpSs
<http://bit.ly/mb3-d65>
M. Garcia, J. Labarta, and J. Corbalan, “Hints to improve automatic load balancing with LeWI for hybrid applications,” Journal of Parallel and Distributed Computing, vol. 74, no. 9, pp. 2781–2794, Sep. 2014.
- vi. **Large scale architectural simulations**
T. Grass et al., “MUSA: A Multi-level Simulation Approach for Next-Generation HPC Machines,” in SC16: International Conference for High Performance Computing, Networking, Storage and Analysis, 2016, pp. 526–537.
- vii. **Arm Scalable Vector Extension**
A. Rico et al. “ARM HPC Ecosystem and the Reemergence of Vectors: Invited Paper,” in Proceedings of the Computing Frontiers Conference, New York, NY, USA, 2017, pp. 329–334.

Account request for accessing the Mont-Blanc platforms

<https://goo.gl/forms/y9dHgGsGtdbuLhxn1>



Conclusions

The Mont-Blanc project



tests, enables, pushes, promotes



Arm based technology into HPC

A closer look  at Arm ecosystem
showed it ready for HPC 

Use cases studied in Mont-Blanc



will be beneficial for future HPC systems

IP-based business model  opens
innovative technological solutions for new markets 

For more information



Interested in any of the topics presented today? Follow us!



montblanc-project.eu



[@MontBlanc_EU](https://twitter.com/MontBlanc_EU)



filippo.mantovani@bsc.es

We are hiring: bsc.es/join-us

