http://www.montblanc-project.eu

Building supercomputers from commodity embedded chips

Paul Carpenter

Barcelona Supercomputing Center

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration [FP7/2007-2013] under grant agreement n° 288777.

Outline

- **Motivation**

- Mont-Blanc project
- Software stack
- Challenges
- Conclusions

Motivation: more performance (earth science)

- Scientists want ever more performance
  - Don't want to hear about "problems": cost, energy, rewriting code…
- Increasing **resolution** to handle extreme events
  - 1km resolution => 100× to 1,000× performance
- Earth **system models**
  - Biological and chemical models
  - 5× to 20× performance
- Quantifying uncertainty
  - **Ensemble** models
  - 10× to 100× performance
- Investigating climate surprises
  - **Long-term simulations** of past and future
  - 10× to 100× performance

- Seems like earth scientists know how to use at least 10^5 to 10^8 × performance

PRACE, The Scientific Case for High-Performance Computing in Europe, 2012—2020, p.49

The problems … and the opportunity

- Nobody knows how to build a sustainable exaflop supercomputer

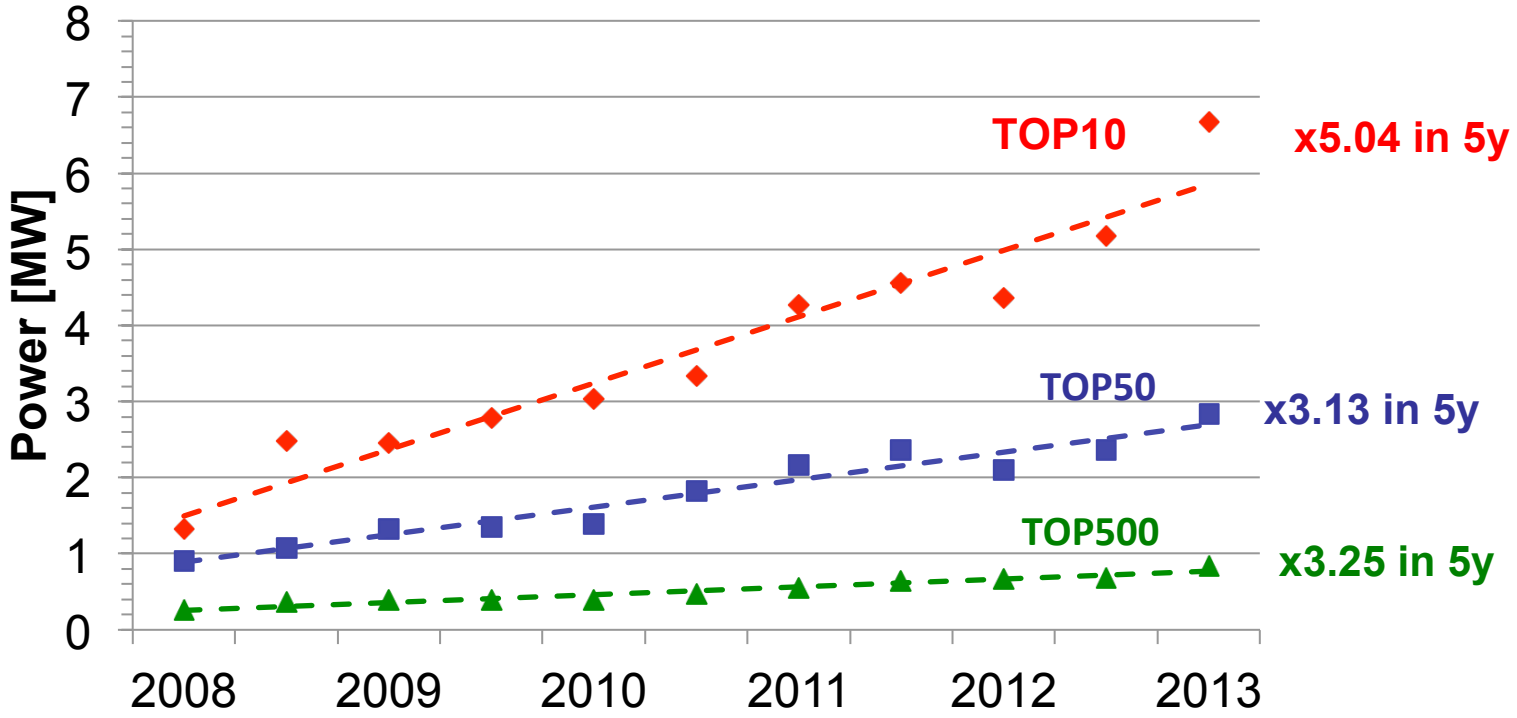
POWER

SPACE

COST

- Europe has little/no HPC technology of its own
  - Yet it represents ~35% of the HPC market
- Europe is strong in embedded computing
  - The most energy- and cost-efficient computing technology today

Problem: TOP500 power consumption evolution



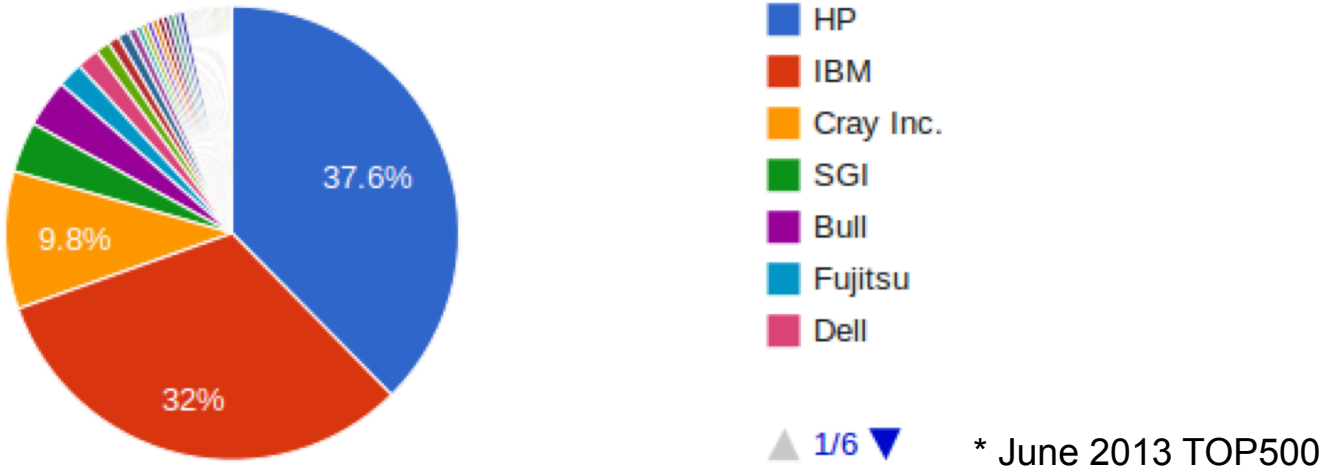
Power excludes cooling

- Performance from higher power consumption
- Is it a problem?
  - LHC fueled by three nuclear power plants (220—330MW)
- Electricity bill: 1MW ~ €1M per year

Graph courtesy of Erich Strohmaier, LBNL

Problem: No European HPC technology

- Europe is about 40% of the HPC market
  - But it has no HPC technology of its own



* June 2013 TOP500
Vendor system share

- Why does it matter?
  - Opportunity for jobs and revenue
  - Reduce dependence on integrated technology from elsewhere
- Significant barriers to exascale computing
  - Consensus is revolutionary approach needed
  - Opportunity for new approaches

Outline

- Motivation

- **Mont-Blanc project**
- Software stack
- Challenges
- Conclusions

Mont-Blanc project

- To develop an **European** Exascale approach
- Leverage **commodity and embedded** power-efficient technology

- Supported by EU FP7 with 16M€ under two projects:
  - Mont-Blanc: October 2011 – September 2014
    - 14.5 M€ budget (8.1 M€ EC contribution), 1095 Person-Month
  - Mont-Blanc 2: October 2013 – September 2016
    - 11.3 M€ budget (8.0 M€ EC contribution), 892 Person-Month

Tibidabo: The first ARM HPC multicore cluster

Q7 Tegra 2
2 x Cortex-A9 @ 1GHz
2 GFLOPS
5 Watts (?)
0.4 GFLOPS / W

Q7 carrier board
2 x Cortex-A9
2 GFLOPS
1 GbE + 100 MbE
7 Watts
0.3 GFLOPS / W

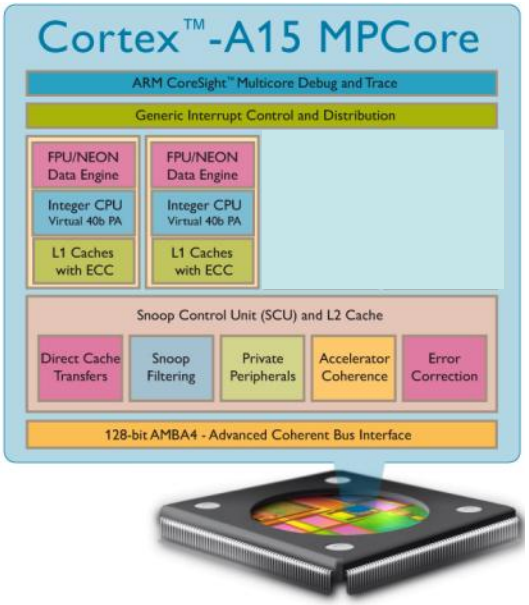
1U Rackable blade
8 nodes
16 GFLOPS
65 Watts
0.25 GFLOPS / W

2 Racks
32 blade containers
256 nodes
512 cores
9x 48-port 1GbE switch

512 GFLOPS
3.4 kWatt
0.15 GFLOPS / W

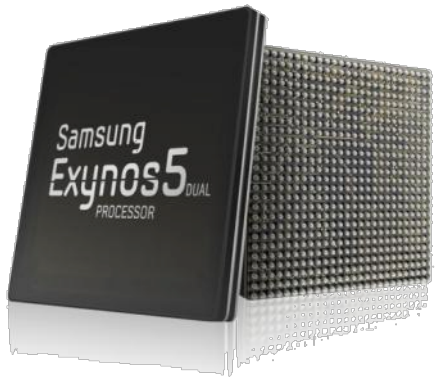
- Proof of concept
  - It is possible to deploy a cluster of smartphone processors
- Enable software stack development

Samsung Exynos 5 Dual Superphone SoC



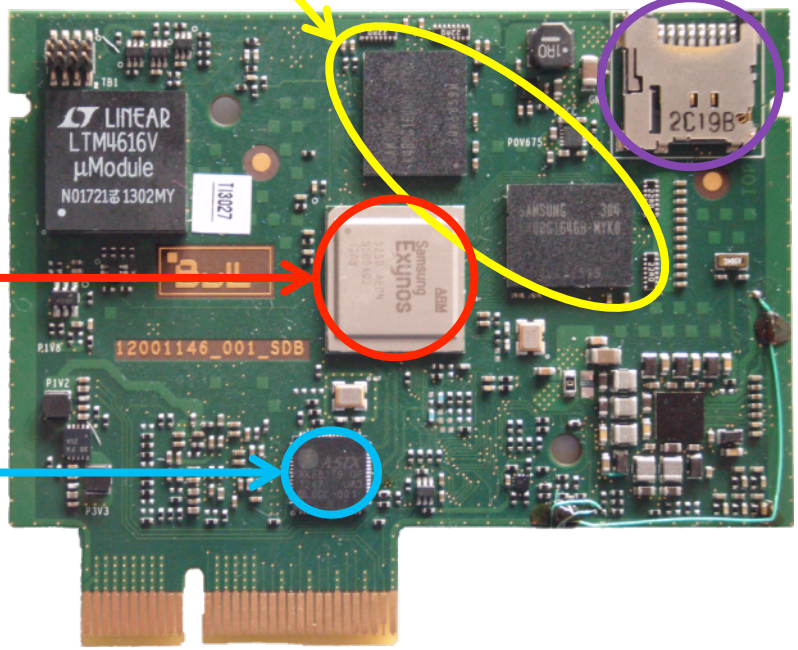
- 32nm HKMG
- Dual-core ARM Cortex-A15 @ 1.7 GHz
- Quad-core ARM Mali T604
  - OpenCL 1.1
- Dual-channel DDR3
- USB 3.0 to 1 GbE bridge

- **All in a low-power mobile socket**



Mont-Blanc SoM

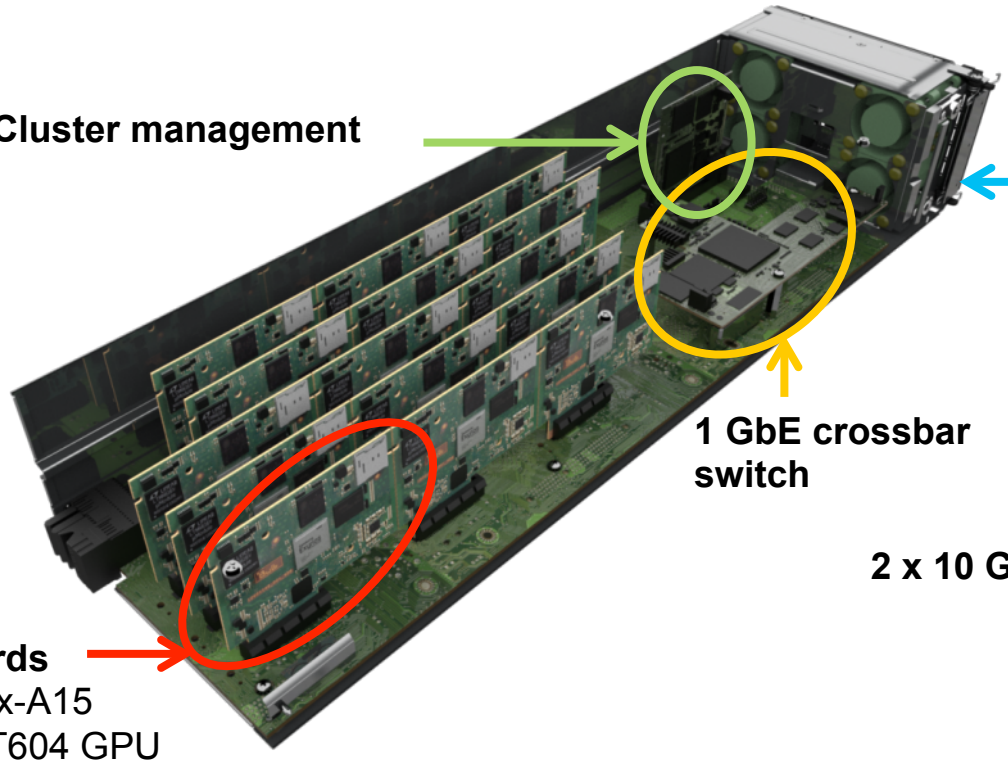
- CPU + GPU + DRAM + storage + network … all in a compute card just 8.5x5.6 cm



6.8GF CPUs
25.5GF GPU
Double precision

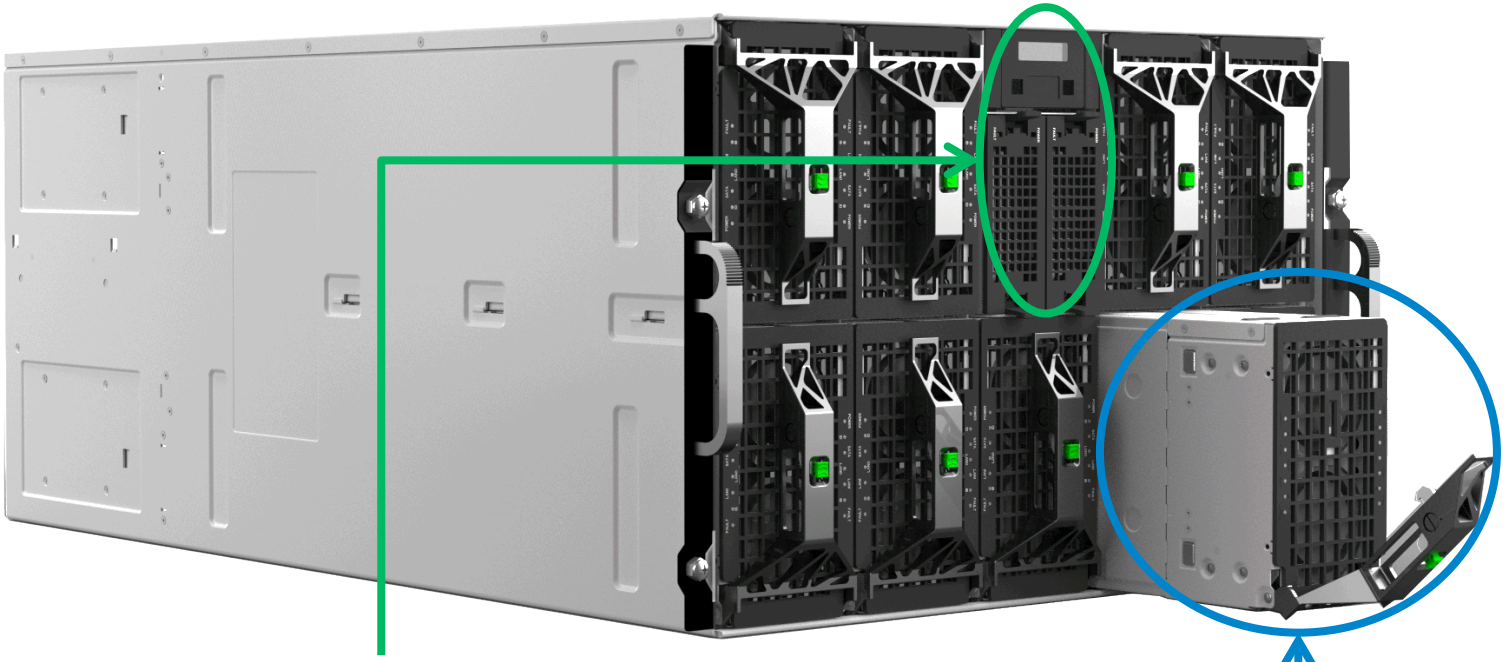
Mont-Blanc server blade

- 15 node-cluster in a standard Bull B505 enclosure



Mont-Blanc server chassis

- 9 blades in a standard 7U BullX chassis
  - Shared cooling, PSU, chassis management

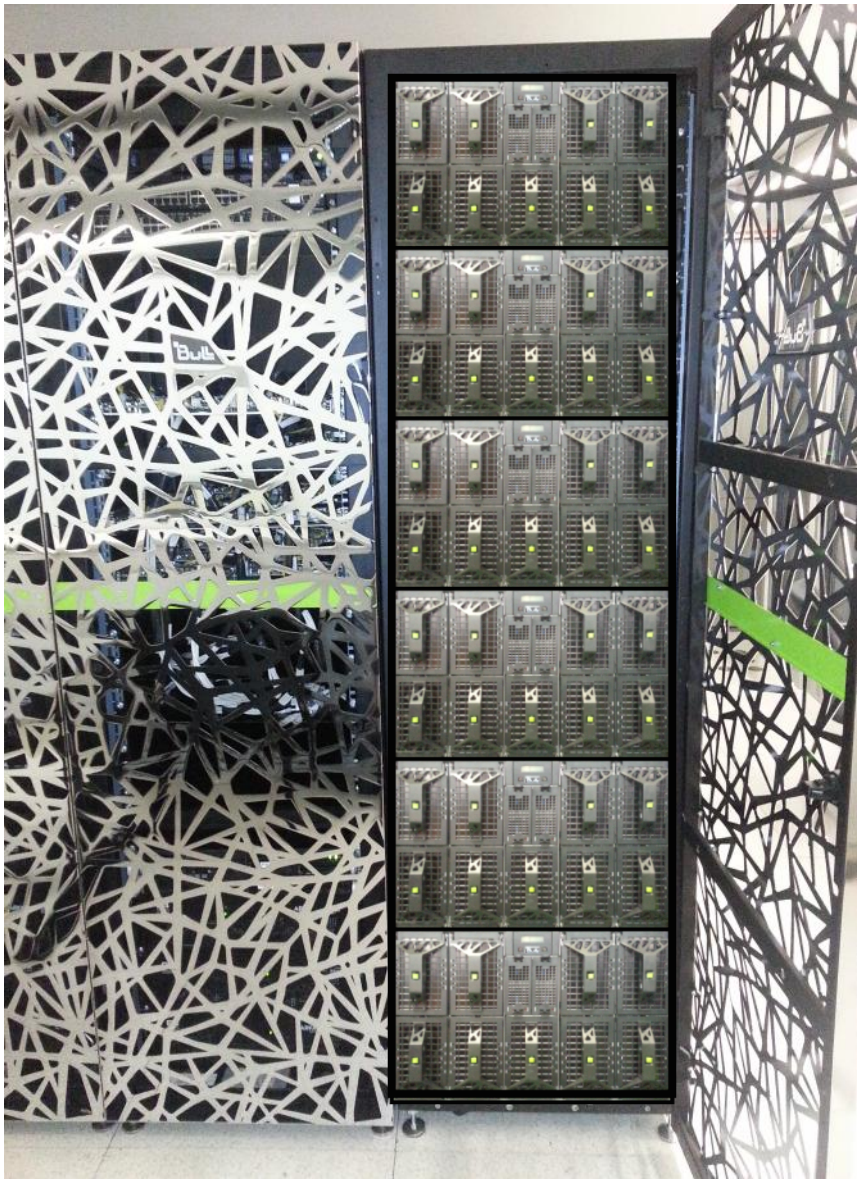


Chassis management panel

9 x Compute blades:
135 x Compute cards + 36 x 10 GbE links
270 x ARM Cortex-A15
135 x ARM Mali-T604 GPU
540 GB DDR3-1600

The Mont-Blanc prototype

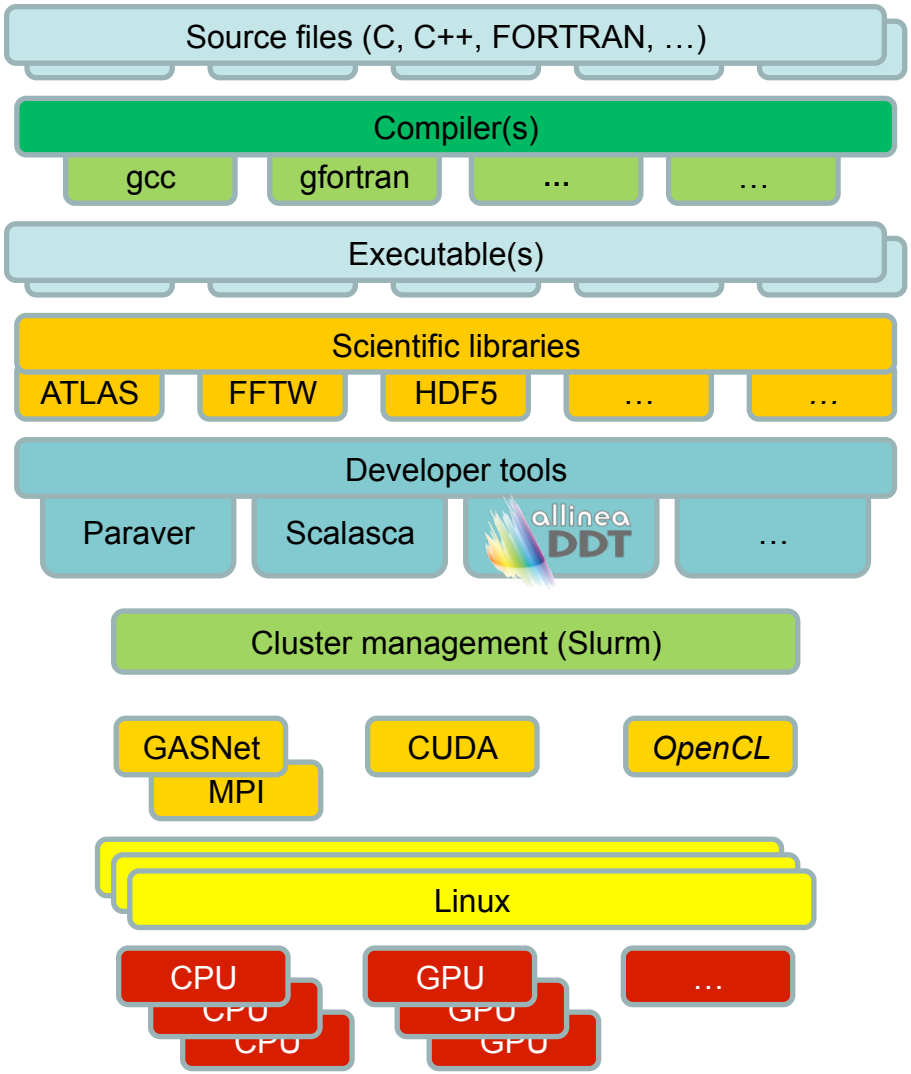
- 6 BullX chassis
- 54 Compute blades
- 810 Compute cards
  - 1620 CPU
  - 810 GPU
  - 3.2 TB of DRAM
  - 52 TB of Flash
- 26 TFLOPS
- 18 kWatt



Outline

- Motivation
- Mont-Blanc project
- **Software stack**
- Challenges
- Conclusions

HPC System software stack on ARM



- Open source system software stack
  - Ubuntu Linux OS
  - GNU compilers
    - gcc, g++, gfortran
  - Scientific libraries
    - ATLAS, FFTW, HDF5,...
  - Slurm cluster management
- Runtime libraries
  - MPICH2, OpenMPI, Nanos++ (OmpSs)
- Performance analysis tools
  - Paraver, Scalasca
- Allinea DDT 3.1 debugger
  - Ported to ARM

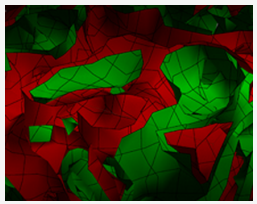
Mont-Blanc micro-kernels

- **All** ported to serial, pthreads, OpenMP, OmpSs, CUDA, OpenCL

Benchmark	Properties
Vector Operation (vecop)	Common operation in regular codes
Dense Matrix-Matrix Multiplication (dmmm)	Data reuse and compute performance
3D stencil (3dstc)	Strided memory accesses (7-point 3D stencil)
2D Convolution (2dcon)	Spatial locality
Fast Fourier Transform (fft)	Peak floating-point, variable-stride accesses
Reduction (red)	Varying levels of parallelism (Scalar sum)
Histogram (hist)	Local privatisation and reduction
Merge Sort (msort)	Barrier synchronisation
N-Body (nbody)	Irregular memory accesses
Atomic Monte-Carlo Dynamics (amcd)	Embarrassingly parallel: compute performance
Sparse Vector-Matrix Multiplication (spwm)	Load imbalance

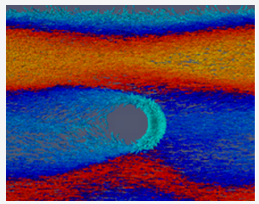
Eleven Mont-Blanc HPC applications

QCD



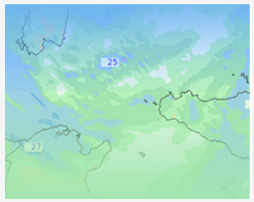
BQCD

Multi-particle collisions



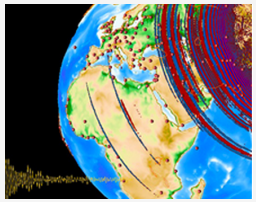
MP2C

Meteorological



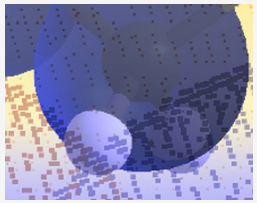
COSMO

Seismic wave

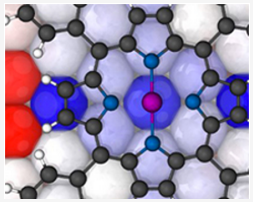


SPECFEM3D

Electronic structure

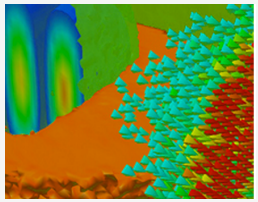


BigDFT



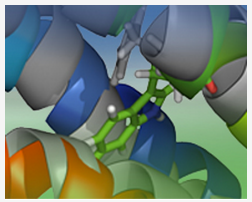
QUANTUM ESPRESSO

Coulomb forces

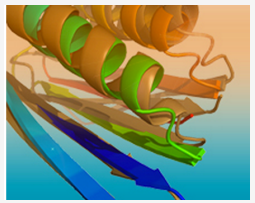


PEPC

Protein folding

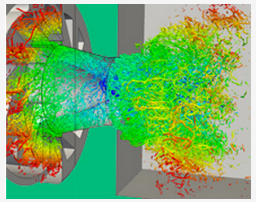


SMMP

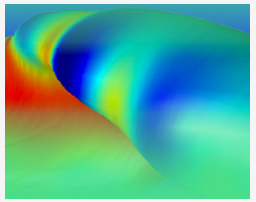


PROFASI

Fluid dynamics

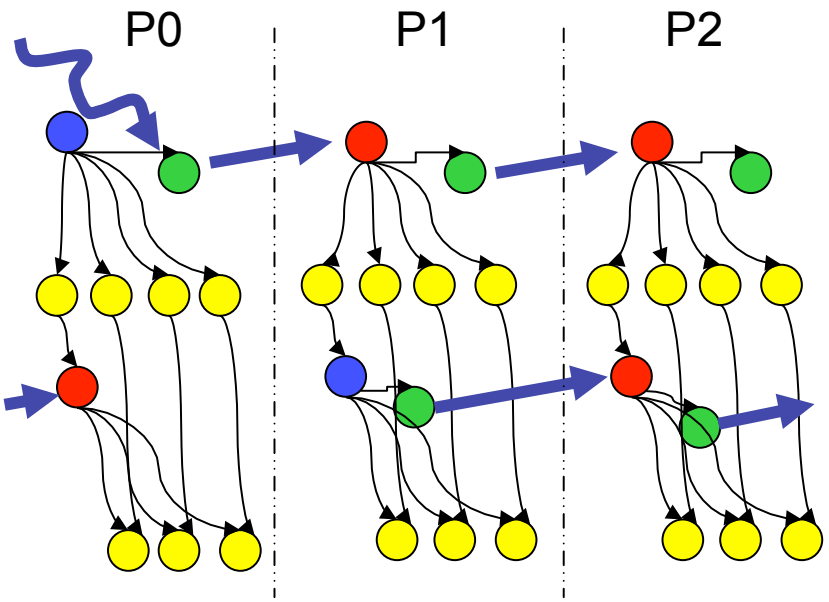


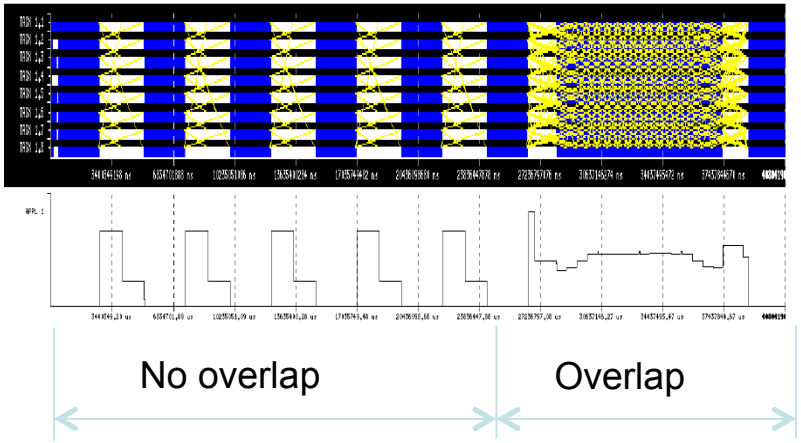
YALES2



EUTERPE

OmpSs runtime layer manages architecture complexity





- Programmer exposed a simple architecture
- Task graph provides lookahead
  - Exploit knowledge about the future
- Automatically handle all of the architecture challenges
  - Strong scalability
  - Multiple address spaces
  - Low cache size
  - Low interconnect bandwidth
- Enjoy the positive aspects
  - Energy efficiency
  - Low cost

Outline

- Motivation

- Mont-Blanc project
- Software stack
- **Challenges**
- Conclusions

Limitations of current mobile processors for HPC

- 32-bit memory controller
  - ARM Cortex-A15 offers 40-bit address space per node
  - But only 4GB per process
- No ECC protection in memory
  - Limited scalability, errors will appear beyond a certain number of nodes
- No standard server I/O interfaces
  - Do NOT provide native Ethernet or PCI Express
  - Provide USB 3.0 and SATA (required for tablets)
- No network protocol off-load engine
  - TCP/IP, OpenMX, USB protocol stacks run on the CPU
- Thermal package not designed for sustained full-power operation

- **All are implementation decisions, not unsolvable problems**
  - Only need a business case to justify the cost of including the new features … such as the HPC and server markets

Further reading

- Rajovic et al, **Supercomputing with Commodity CPUs: Are Mobile SoCs Ready for HPC?** SC13, Denver, CO
  - Best student paper award
  - (I have some printed copies with me)

Conclusions

- HPC is key to industry and research
- Need sustainable EFLOPS technology
- Exascale computing requires revolutionary technology
- Convergence of embedded and HPC
  - Embedded cores and accelerators capable of running HPC
  - Integration is key to energy efficiency
    - SoC containing CPU + GPU + NIC + NoC + memory
- Leverage European strengths
  - Embedded computing
  - Parallel programming models

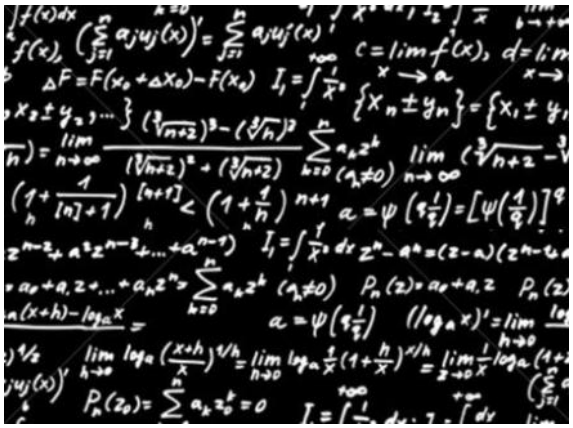

montblanc-project.eu


MontBlancEU


@MontBlanc_EU

BACKUP SLIDES

Motivation: The need for supercomputers



- Supercomputers are a basic research tool
  - Societal challenges: diseases, human brain, climate, catastrophes
  - Science: astrophysics, particle physics
  - Industry: oil, gas, innovation in products and services
- Named by many as the *“third pillar of science”*
- Scientists want ever more performance
  - Don’t want to hear about “problems”: cost, energy, rewriting code…

First, vector processors dominated HPC

- 1st TOP500 list (June 1993) dominated by DLP architectures
  - Cray vector,41%
  - MasPar SIMD, 11%
  - Convex/HP vector, 5%
- Fujitsu *Wind Tunnel* is #1 1993-1996, with 170 GFLOPS

Then, commodity took over special purpose

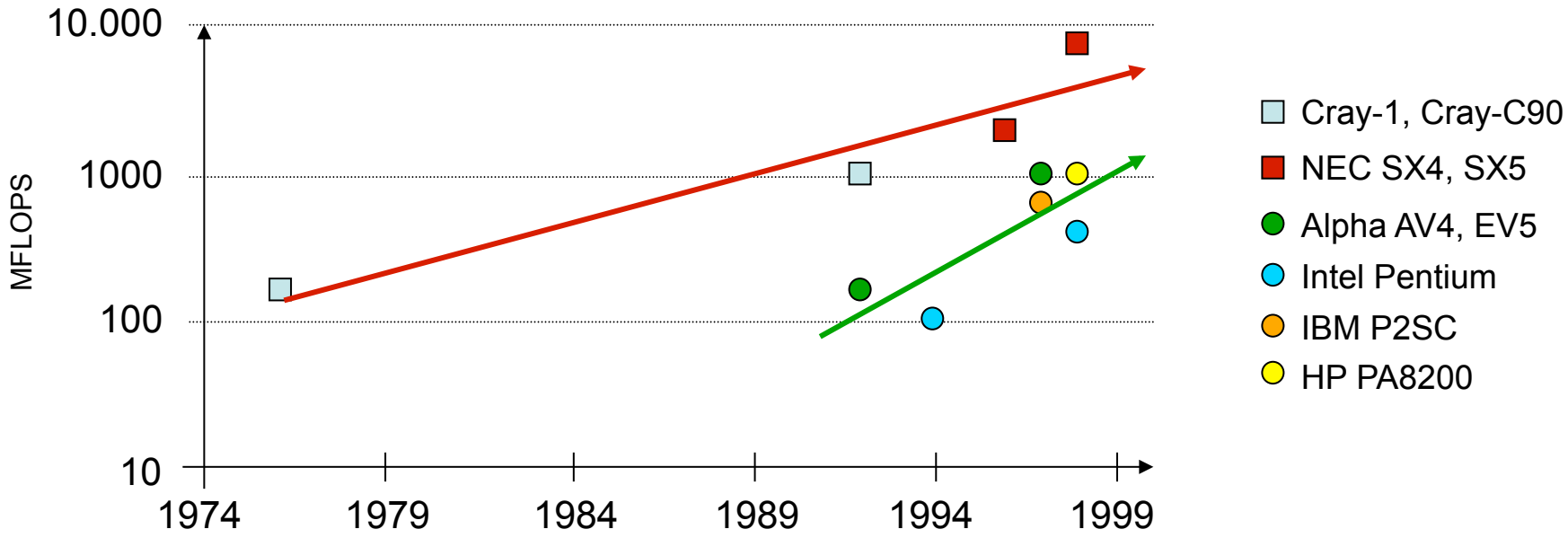
- ASCI Red, Sandia
  - 1997, 1 TFLOPS
  - 9,298 cores @ 200 MHz
  - Intel Pentium Pro
    - 1999: upgraded to Pentium II Xeon, 3.1TF

- ASCI White, LLNL
  - 2001, 7.3 TFLOPS
  - 8,192 proc. @ 375 MHz,
  - IBM Power 3

Benefits so compelling that software was rewritten..

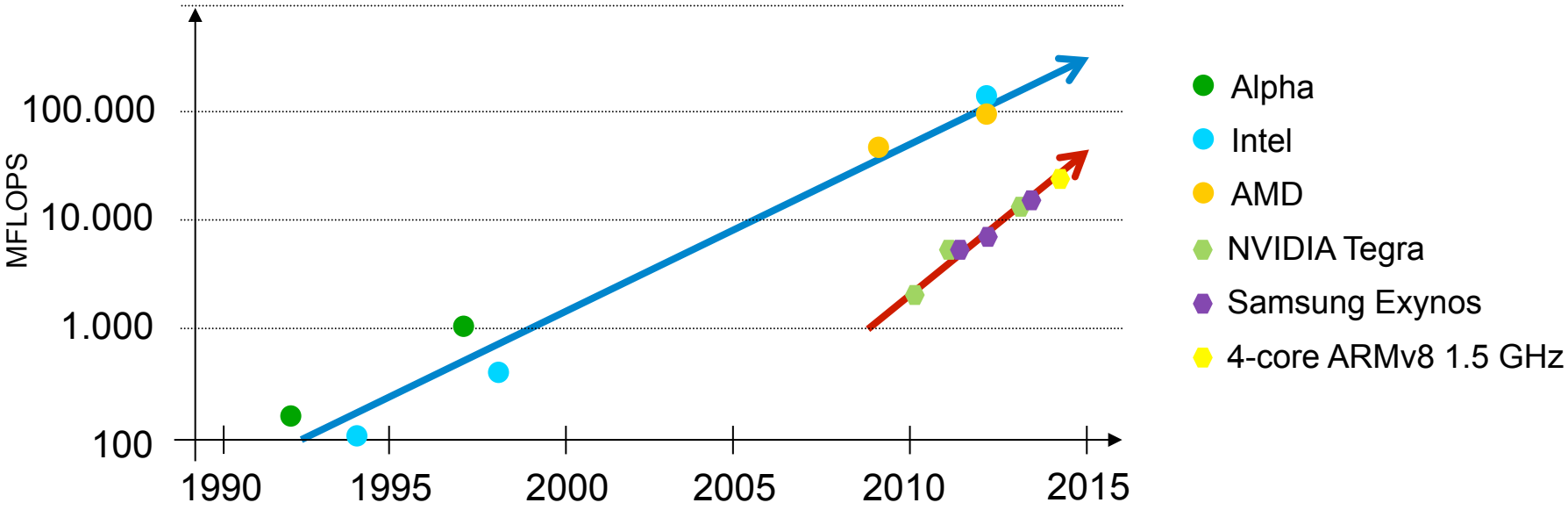
Transition from vector parallelism to message-passing programming models

The killer microprocessors



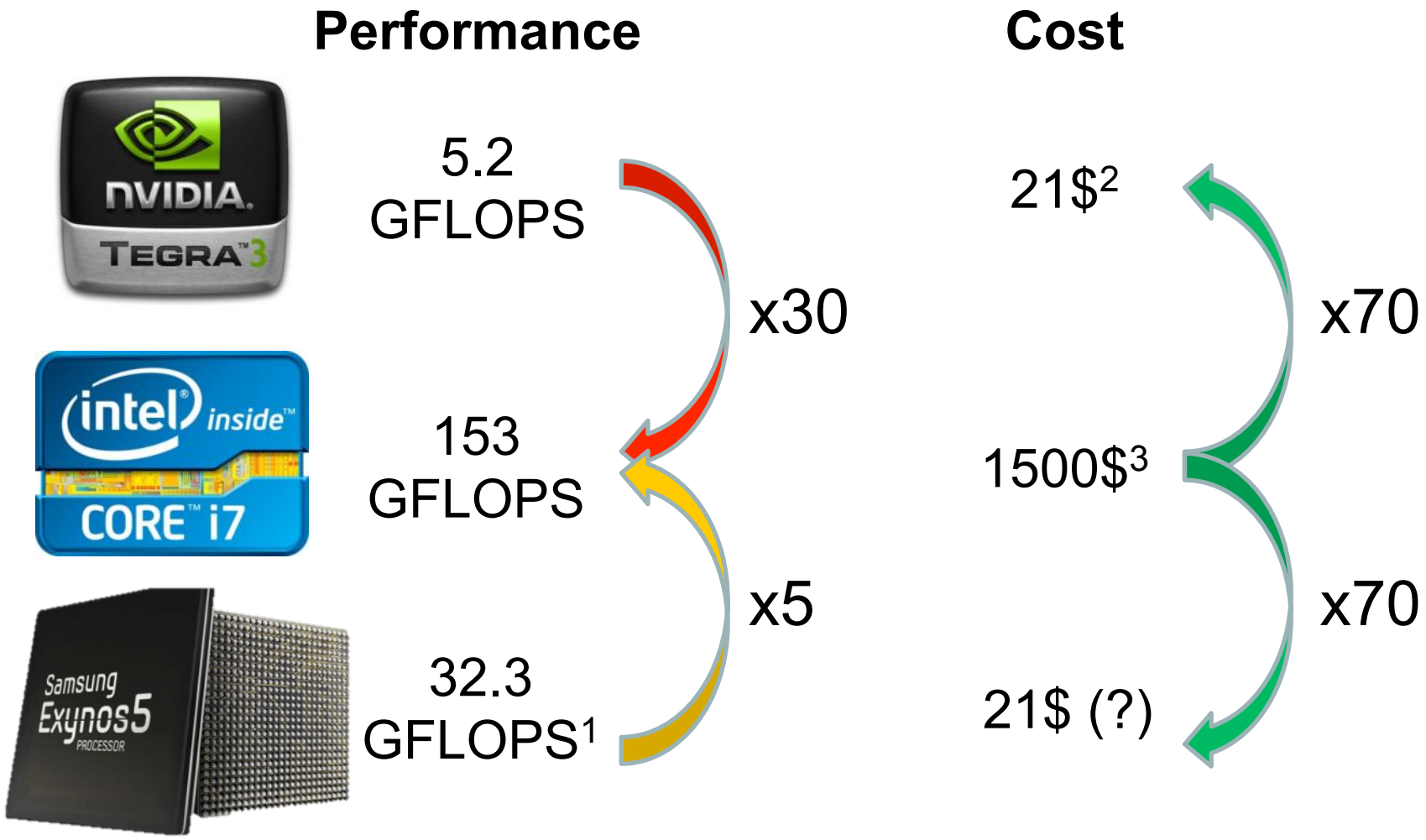
- Microprocessors killed the Vector supercomputers
  - They were not faster ...
  - ... but they were significantly cheaper and greener
- Need 10 microprocessors to achieve the performance of 1 Vector CPU
  - SIMD vs. MIMD programming paradigms

The killer mobile processors™



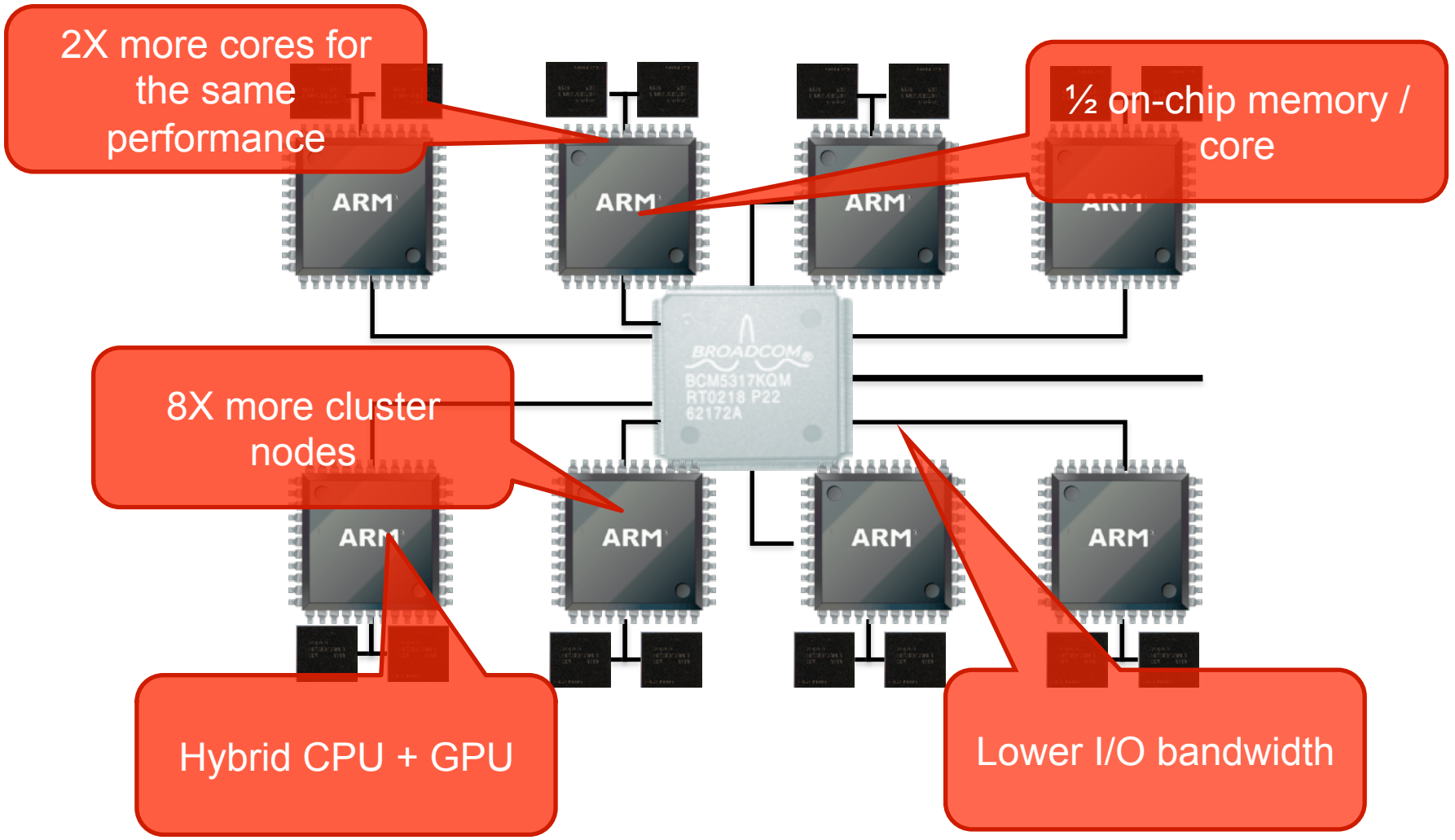
- History may be about to repeat itself …
  - Mobile processor are not faster …
  - … but they are significantly cheaper
- Need 10 smartphones to achieve the performance of 1 High Performance CPU
  - Leverage the embedded GPU accelerator

Mobile SoC vs Server processor



1. 6.8 GFLOPS from CPU + 25.5 GFLOPS from embedded GPU
2. Leaked Tegra3 price from the Nexus 7 Bill of Materials
3. Non-discounted List Price for the 8-core Intel E5 SandyBridge

There is no free lunch

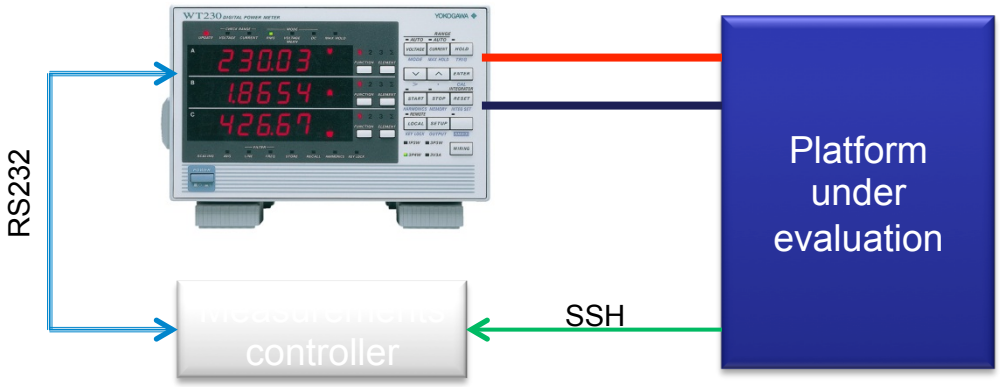


Get ready for the change, before it happens …

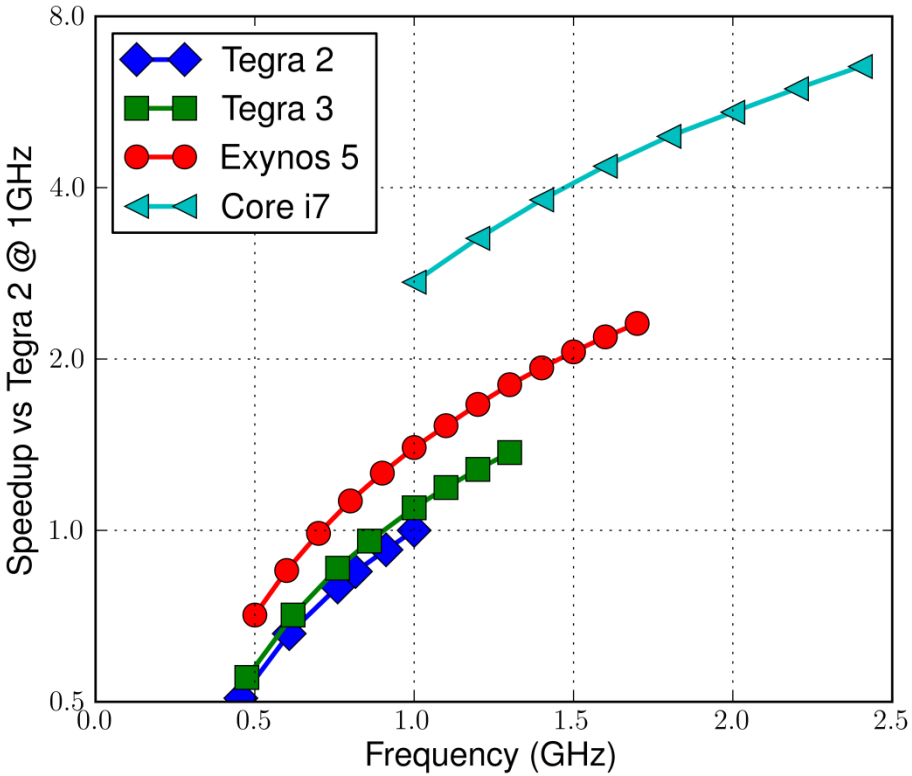
- Mobile processors have qualities that make them interesting for HPC
  - FP64 capability
  - Performance increasing rapidly + energy efficient
  - Embedded GPU accelerator
  - Large market, many providers, competition, low cost
- Current limitations are due to target market conditions
  - Not real technical challenges
- A whole set of ARM server chips is coming
  - Solving most of the limitations identified

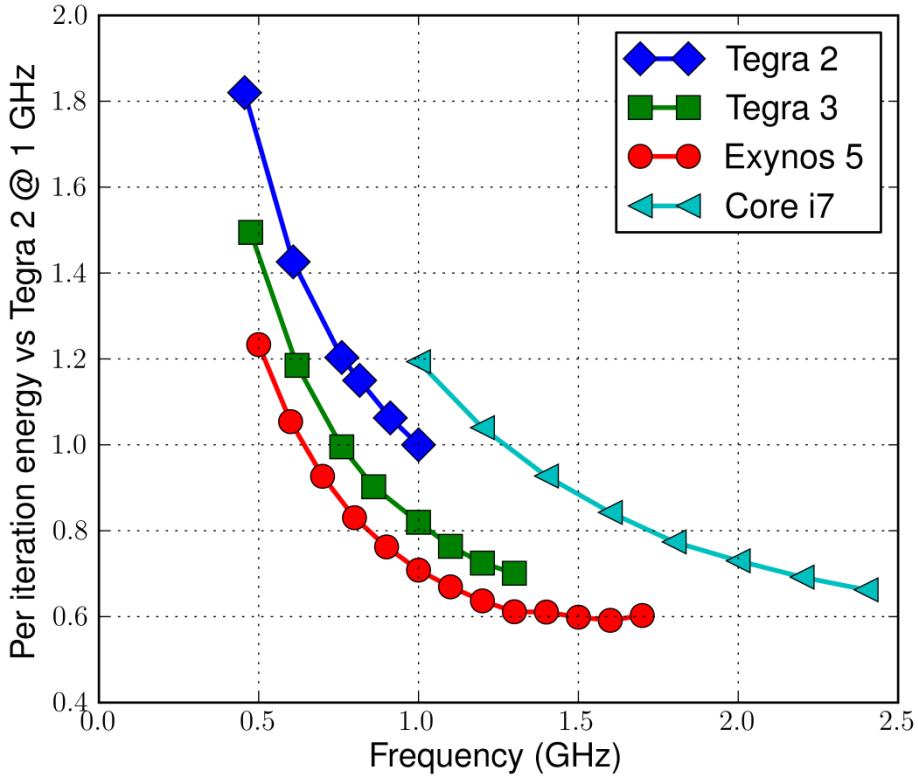
Methodology

- Same input sets for all platforms
- Power samples collected in compute regions
- Number of iterations maintains same total execution time on all platforms
  - Long enough to minimize synchronization effects of power measurement



Single core performance and energy

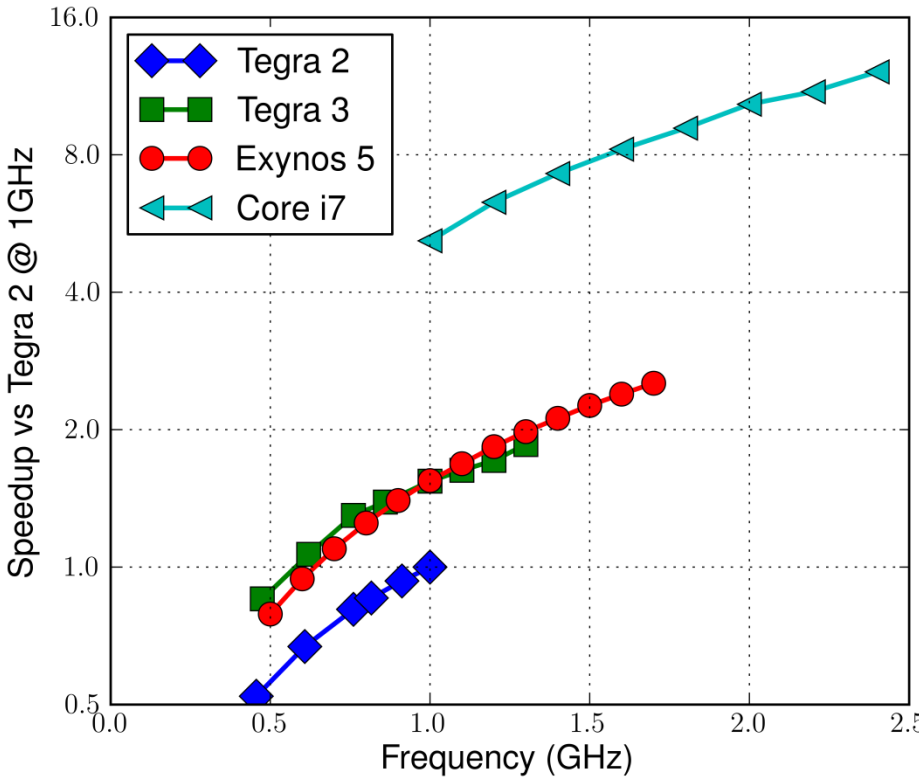


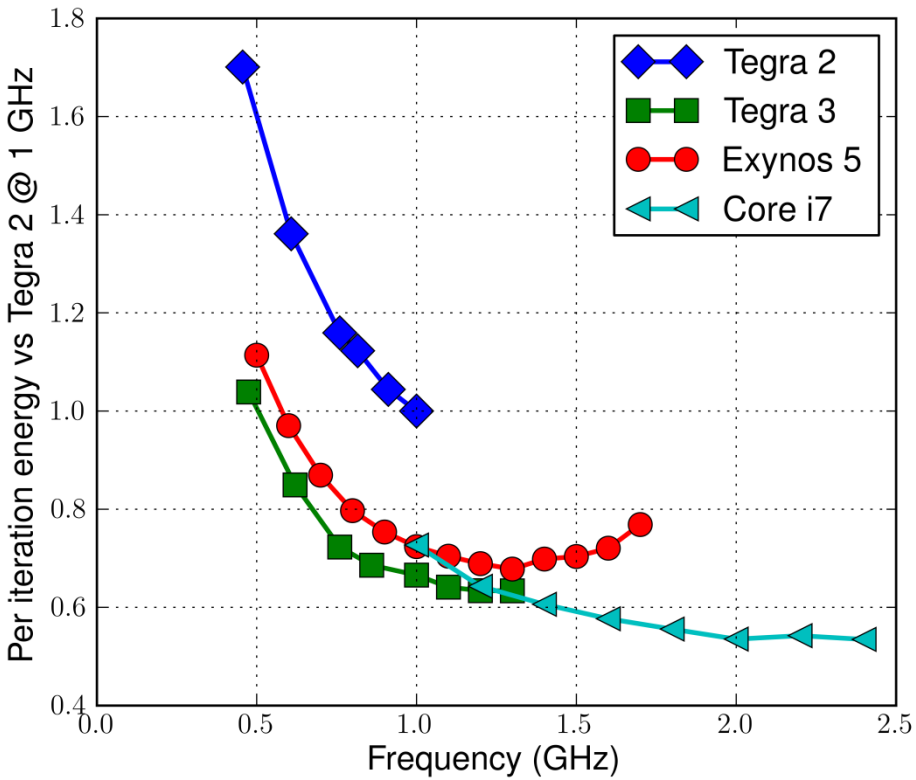


- Cortex-A9 in Tegra3 is 1.4x faster than Tegra2 (higher clock frequency)
- Cortex-A15 in Exynos5 is 1.7x faster than Cortex-A9 in Tegra3
  - Higher clock frequency, higher memory bandwidth, and better core microarchitecture
- Core i7 is ~3x faster than Cortex-A15 in Exynos5 at maximum frequency
  - 2x faster at the same frequency

- Mobile SoC platforms as efficient as Core i7 platform at their highest operating points

Multicore performance and energy

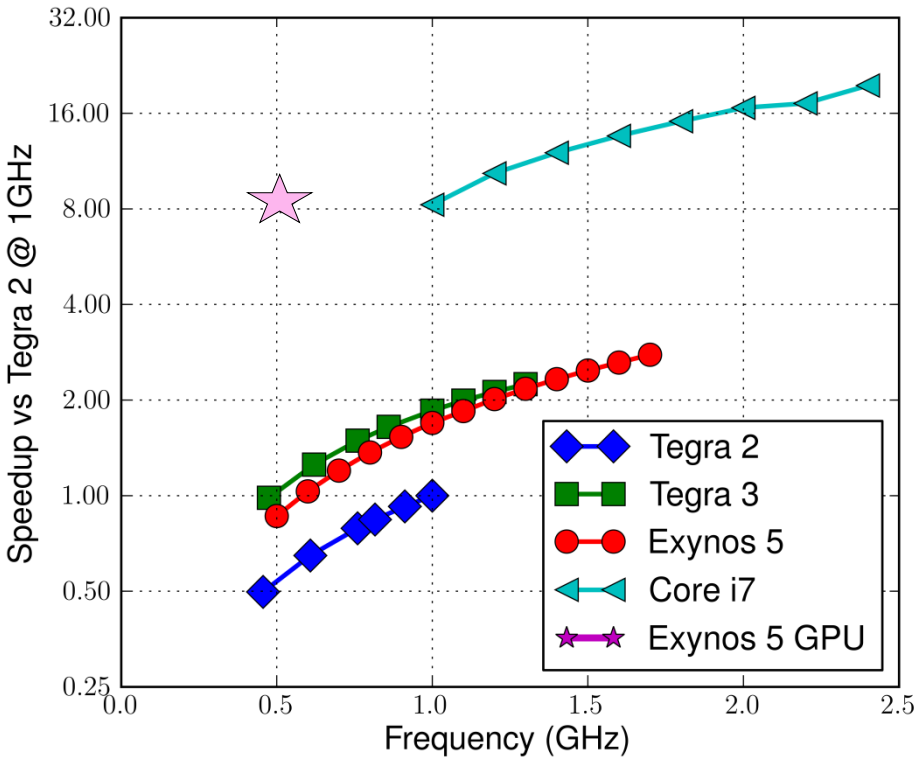


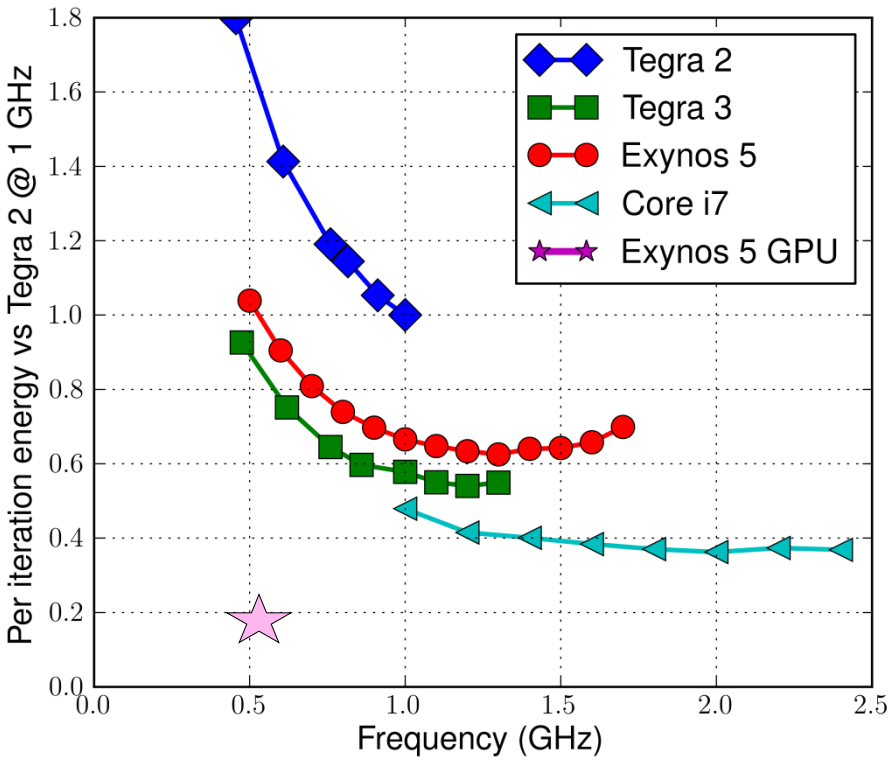


- Tegra3 platform as fast as Exynos5 platform, a bit more energy efficient
  - 4-core Cortex-A9 vs. 2-core Cortex-A15
- Corei7 is 6x faster than Exynos5 at maximum frequency

- Tegra3 and Exynos5 as efficient as Corei7 at the same frequency

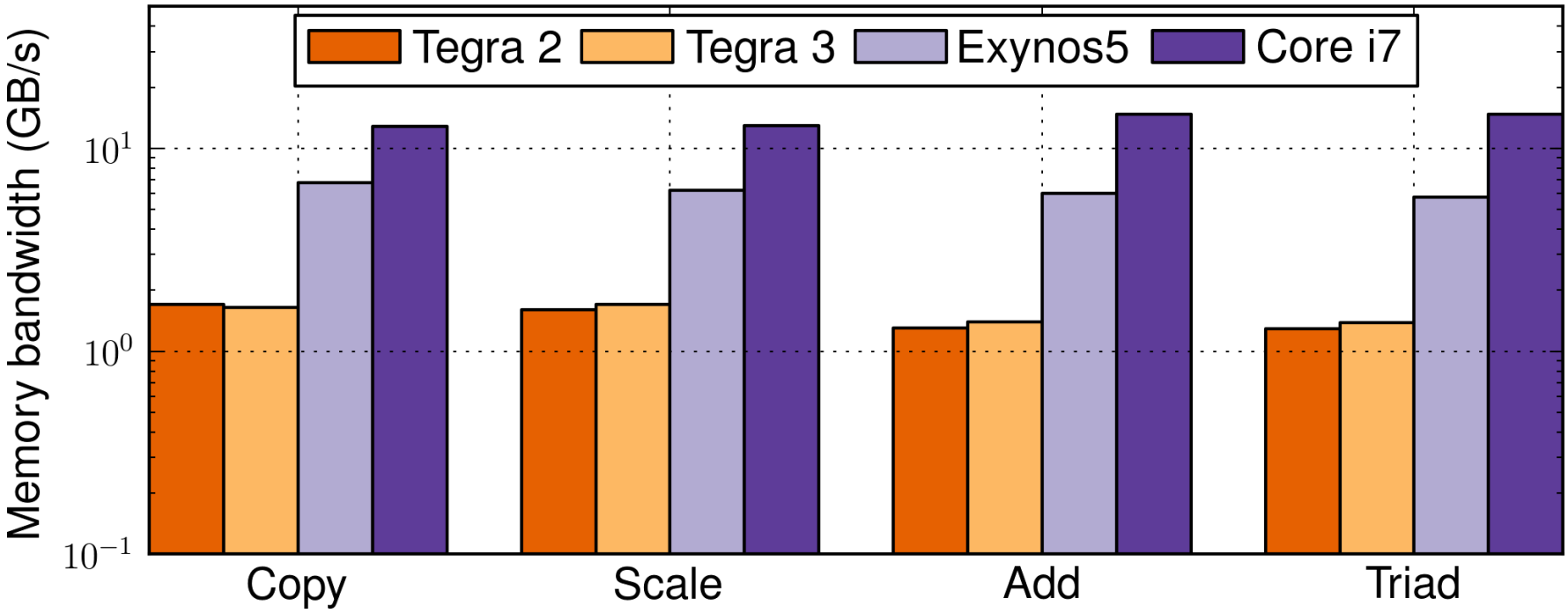
Addendum – Mobile SoCs GPU vs. multicore





- Exynos 5 also integrates a compute capable Mali-T604 GPU
- Exynos 5 GPU platform as fast as Core i7 platform at 1GHz
  - 3 times faster than ARM Cortex-A15 dual core at max. frequency
- Exynos 5 platform becomes the most energy efficient

Memory bandwidth (STREAM)



- Exynos 5 improves dramatically over Tegra (4.5x)
  - Dual-channel DDR3
  - ARM Cortex-A15 sustains more in-flight cache misses
- Core i7 provides ~2x more memory bandwidth than Exynos5