Measurements of the LHCb software stack on the ARM architecture

Marco Clemencic, Ben Couturier, Niko Neufeld, Vijay Kartik Subbiah

Physics Department CERN CH-1211 Geneva 23, Switzerland

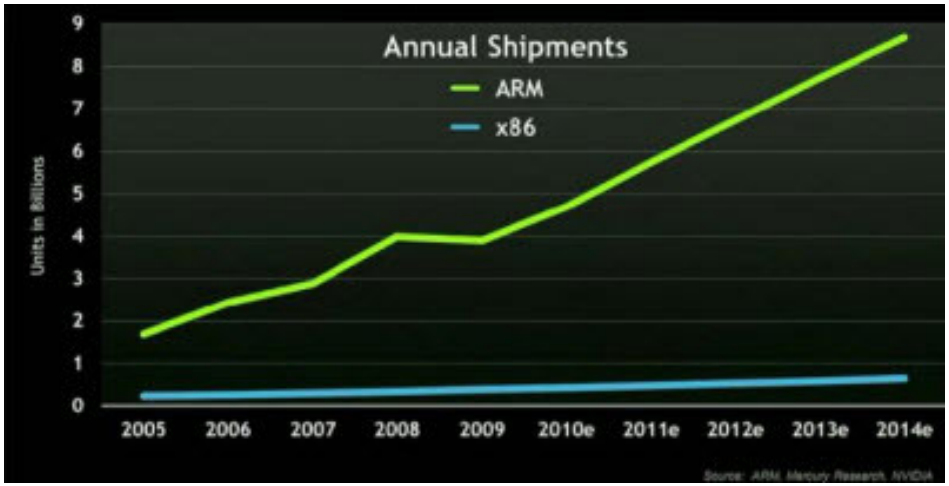
Computing in High Energy and Nuclear Physics, 2013

Overview

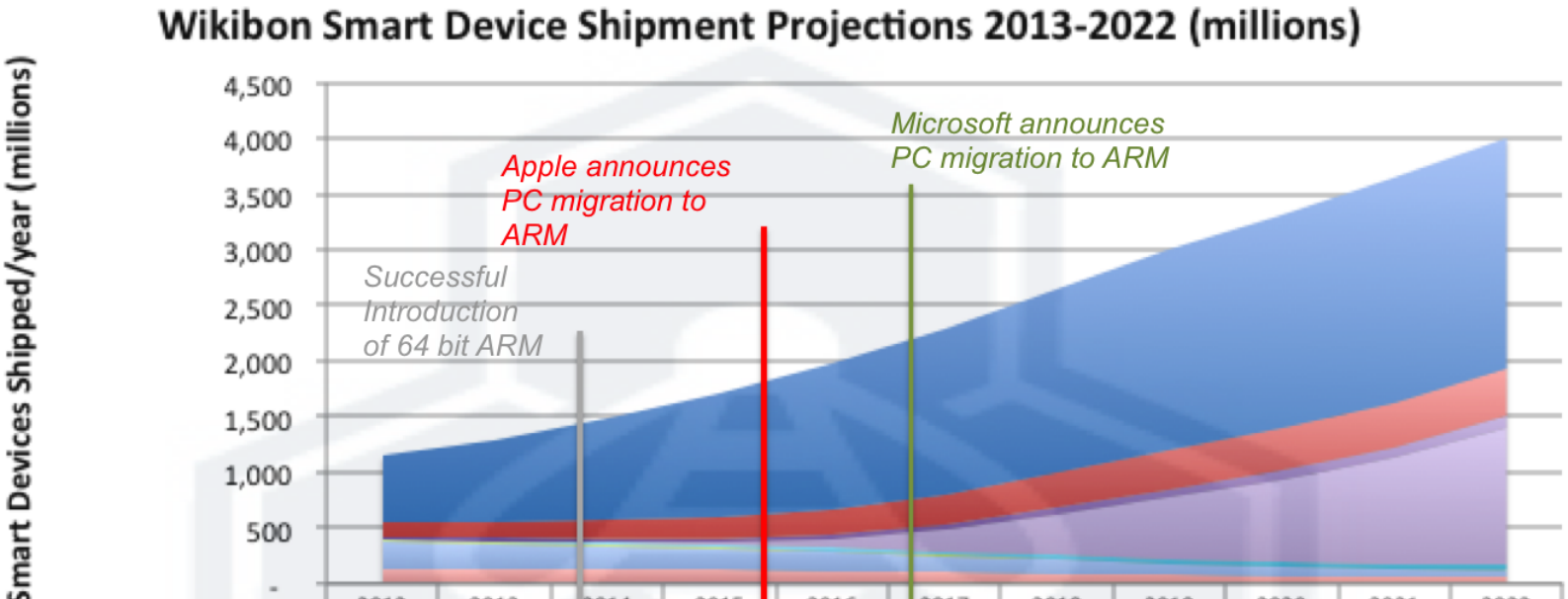
1. Why ARM?
2. In practice
3. Measurements
4. Odds and ends

The Market

- Most tablets and smart-phones powered by ARM processors
- 2013 will see more shipments of tablets than laptops
- Server market has virtually no growth, desktops declining



Nightmare in Santa Clara

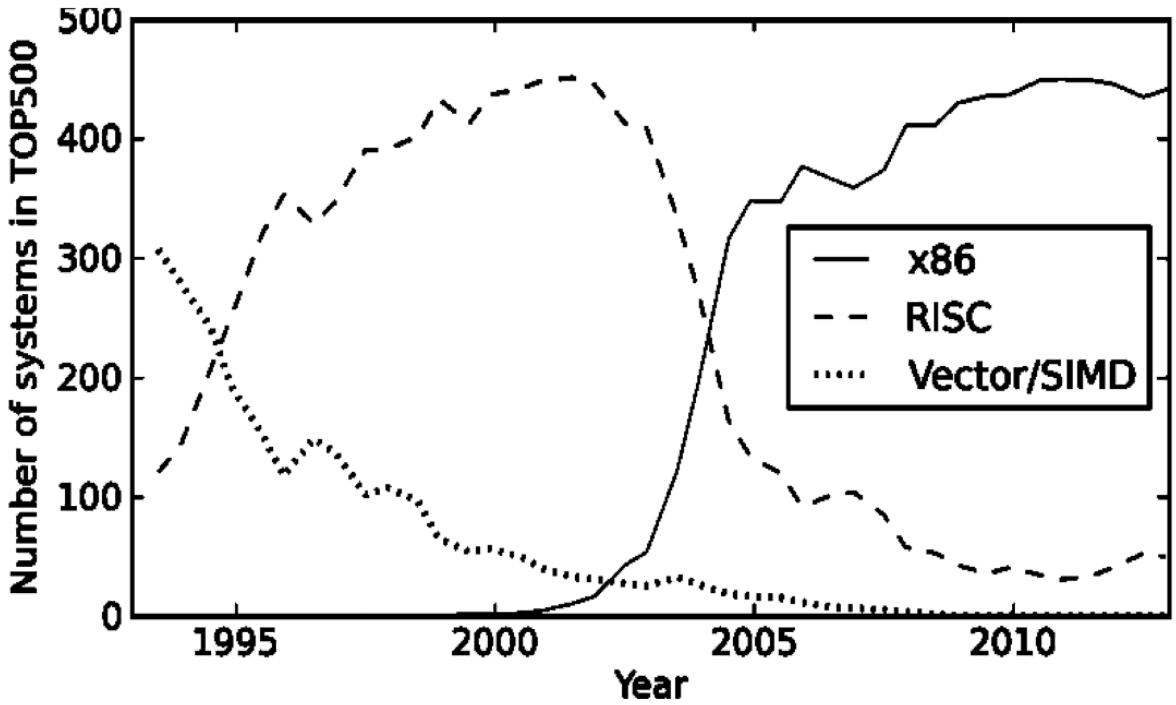


	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022
ARM Smartphones	604	743	906	1,097	1,305	1,488	1,651	1,800	1,926	2,022	2,083
ARM Tablets	121	142	168	198	232	269	309	346	377	400	412
ARM PCs, Chromebooks, etc.	22	25	29	34	39	46	55	70	85	100	115
ARM Smart Wearable Devices	-	-	8	25	74	186	371	557	724	941	1,223
Other ARM Devices	15	18	21	25	29	34	38	43	47	51	54
eReaders	20	15	12	9	7	6	5	5	4	4	3
x86 Laptop PCs	229	217	204	188	167	144	119	96	79	68	61
x86 Desktop PCs, Workstations	137	132	127	120	111	100	88	75	65	57	51
x86 Tablets	2	3	4	6	7	9	9	8	6	4	3
Ratio ARM/x86	2.1	2.7	3.4	4.4	5.9	8.0	11.3	15.8	21.0	27.2	34.0

Year

Source: Wikibon 2013, IDC & Gartner 2012 shipments & Wikibon 2013-2022 projections. Assumption: Apple & Microsoft migrate to successful 64-bit ARM.

So you believe that Vax, Alpha, MIPS, ***your favourite architecture here***, are eternal?



This figure taken from a presentation by the Mont Blanc project *

- μ-processors weren't faster
- They were much cheaper and "greener"

- Mobile processors are not faster
- They are much cheaper and greener

*

The computer-farm of the future

Not really. . .

The computer-farm of the future

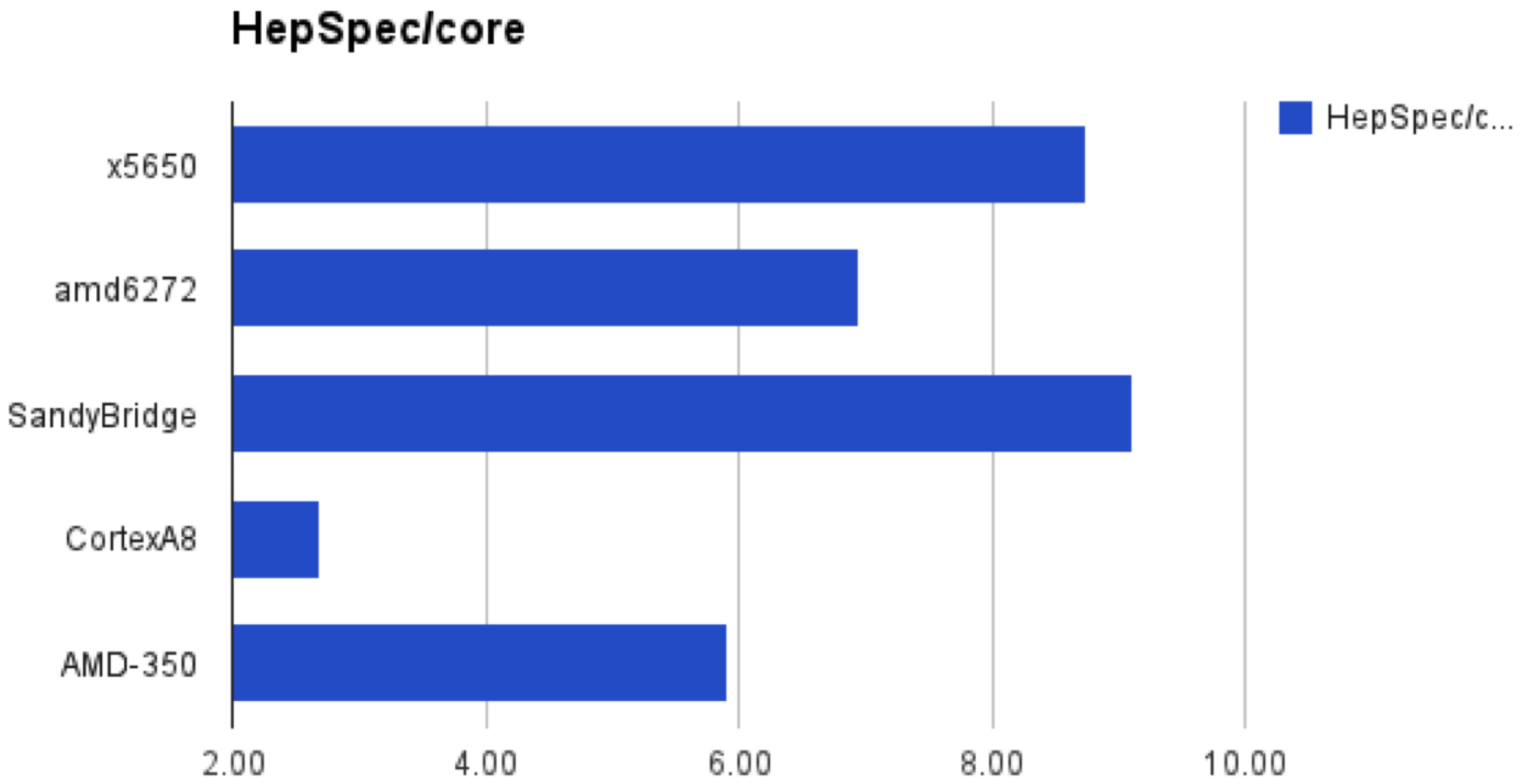
Not really. . .

Micro-servers

- ARM-based SoCs are widely available now
- Lots of interest in micro-servers (cloud, web-shops)
- Still glue-logic is needed (no fast PCIe/SATA on current SoCs)
- Dense packaging of multiple SoCs
- Example (and used in these tests) Boston Viridis (based on Calxeda)

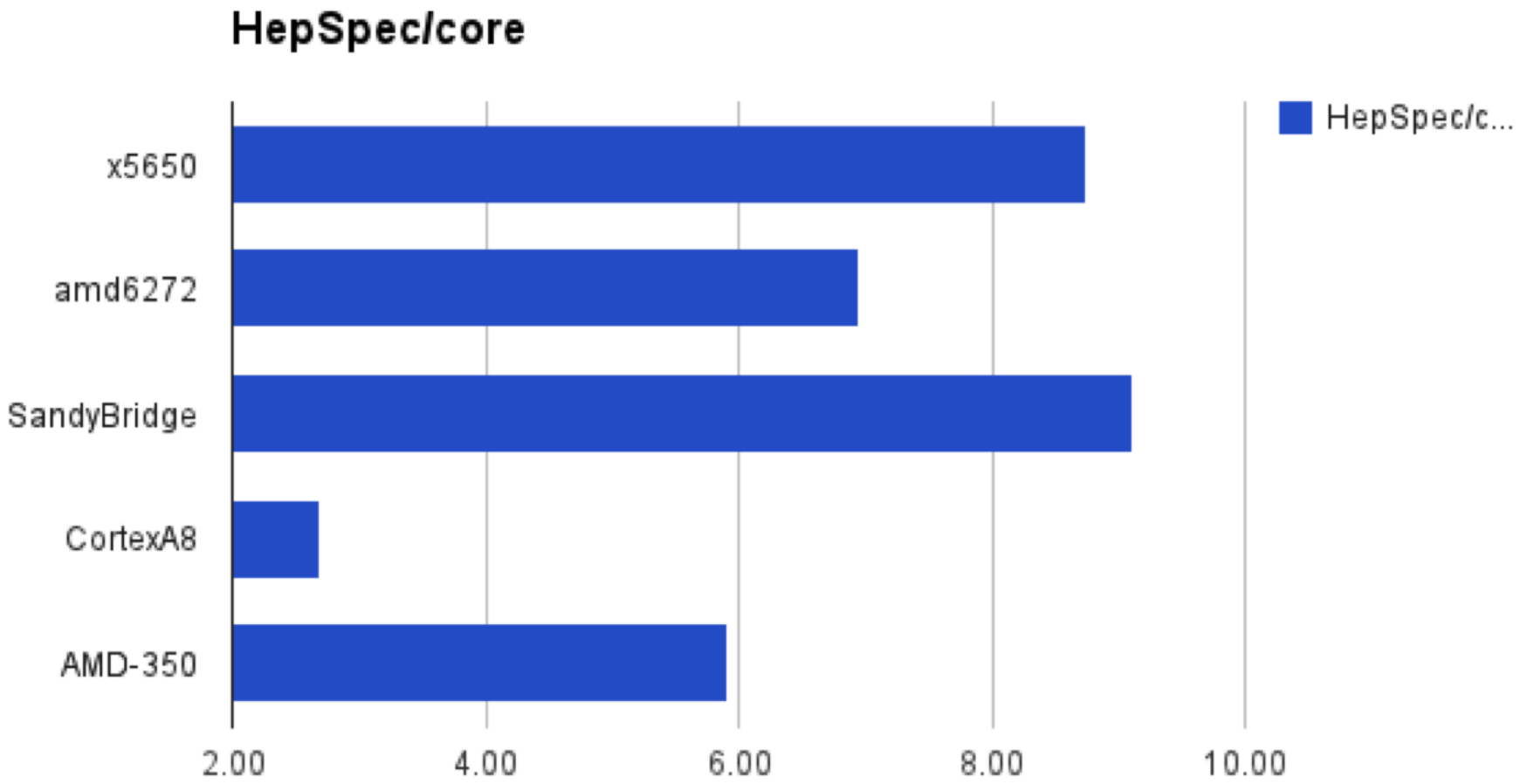
- 48 SoC, 4 cores 4 GB RAM
- ARM A9 Cortex 1.4 GHz (v7 architecture 32-bit)
- 80 Gb Ethernet switch (10 GigE external)
- Total 192 cores / 192 GB RAM /300 Watt
- redundant power, etc. . .

First look: HEPSpec



So we need many

First look: HEPSpec



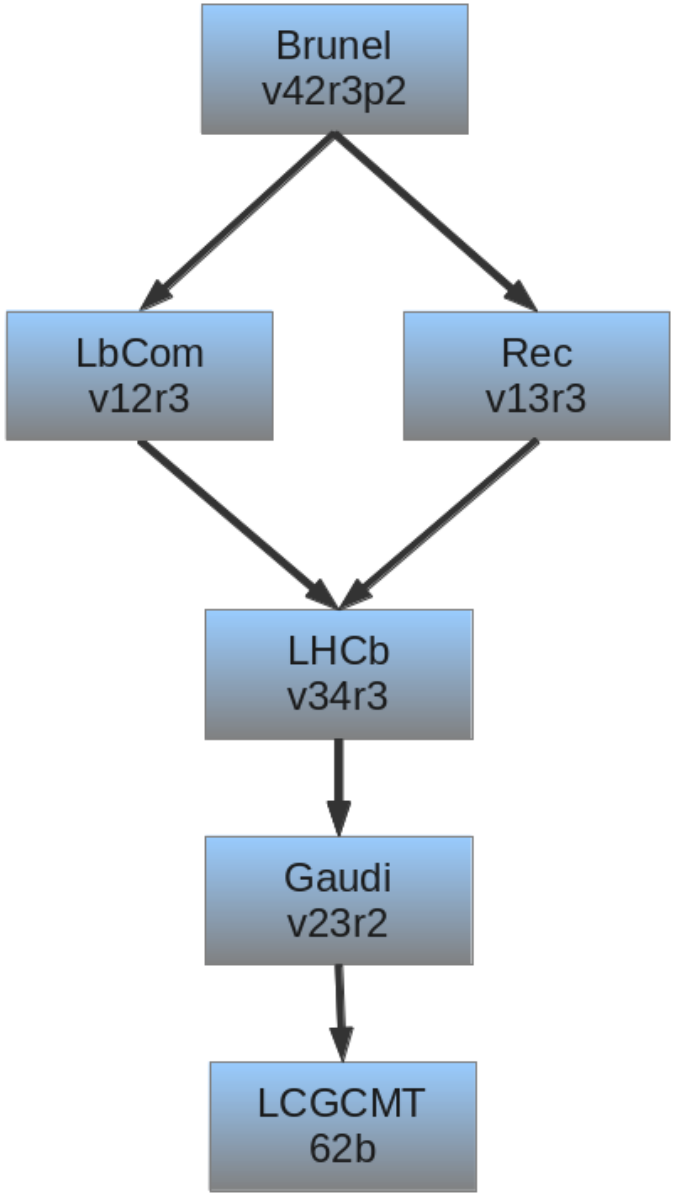
So we need many

HEP spec is not enough - do the real test

- HepSPEC is not necessarily a good test for Online usage
  - Online we (currently) run n instances of the same application in parallel, where n is $\geq$ number of cores/hyperthreads
  - In such a scenario hyperthreading typically adds overproportionally (up to 40% of total machine performance) compared to mixed work-loads
- Need to benchmark a real LHCb work-load: Brunel (the reconstruction program)

The task

- 3.6 MLOC of code (excluding LCG_CMT)
- ROOT v5.34.05
- gcc 4.7.2, Boost 1.51



The platform

- Remote test
  - CPU = Calxeda EnergyCore (SoC), ARM Cortex-A9 CPU, 4 cores, $\approx$ 1.1 GHz, 4GB RAM
  - `Linux cloud12 3.6.10-8.fc18.armv7hl.highbank 1 SMP Tue Jan 29 14:01:38 EST 2013 armv7l armv7l armv7l GNU/Linux`
- Local development, CARMA
  - CPU = NVIDIA Tegra 3, a Quad-core ARM Cortex-A9 CPU $\approx$ 1.3 GHz
  - Ubuntu 11.04
  - `Linux carma-devkit 3.1.10-carma 2 SMP PREEMPT Fri Aug 31 15:28:42 PDT 2012 armv7l armv7l armv7l GNU/Linux`
  - Not a hard-float kernel - (not good)

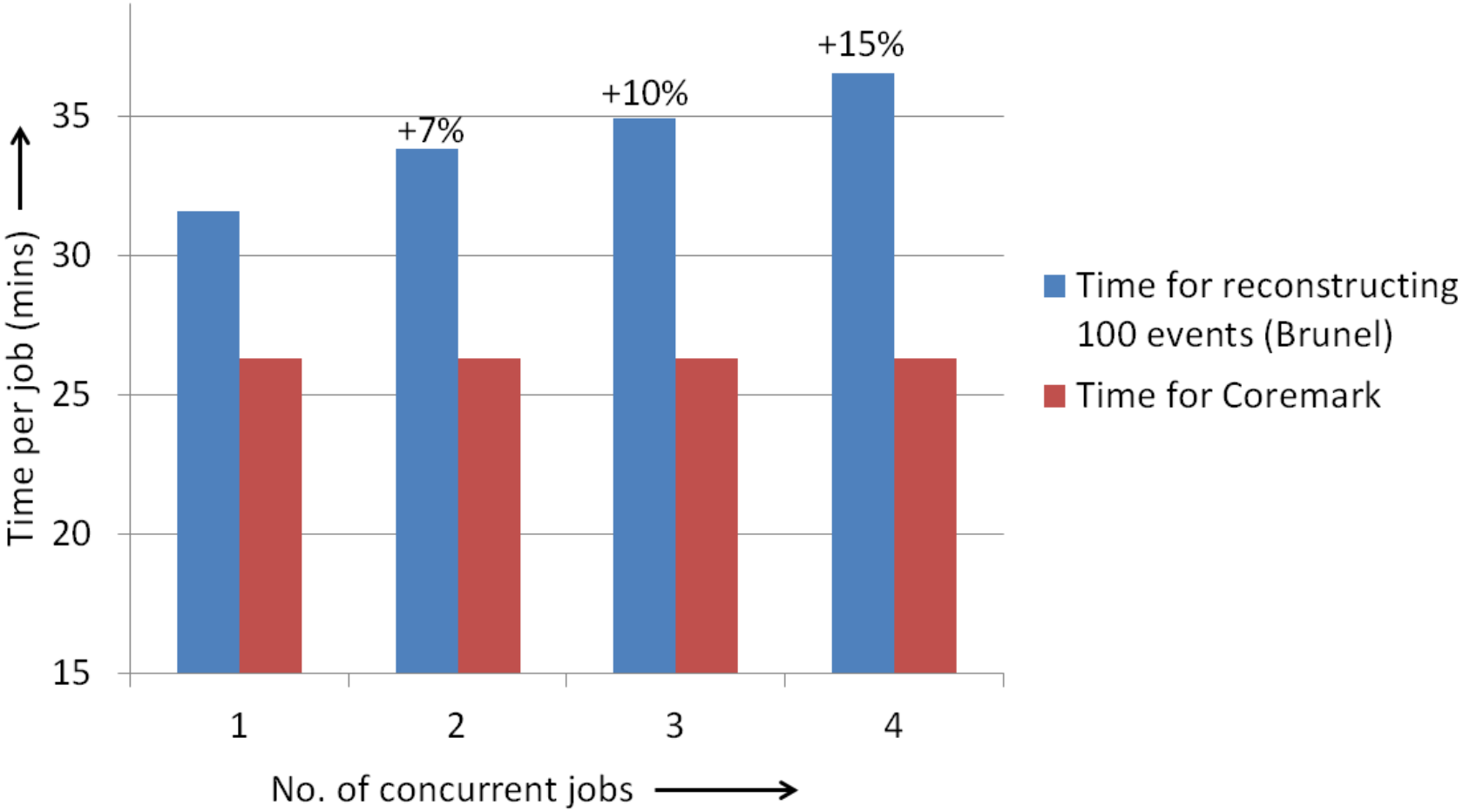
Single core performance

	CARMA	Viridis	x86 L5520
freq. (GHz)	1.3	1.1	2.27
# cores	4	4	8 (with HT)
total RAM (GB)	4	4	48
time (h)	~ 10	~ 2.5	~ 0.5

- Time for reconstructing the same 1000 events in Brunel
- Events read from network share (NFS/AFS)
- Test results stable

Scaling

- Loading the entire machine with identical jobs
- x86 scales **linearly** for real cores, HT add about 40%
- ARM memory bandwidth does not scale



Correctness

"Brem Match"	sum		mean/eff^*		rms/err^*	
	ARMv7	x86_64	ARMv7	x86_64	ARMv7	x86_64
#calos	50085	50085	60.489	60.489	30.140	30.140
#chi2	2.737109e+09	2.737105e+09	5009.1	5009.1	2866.4	2866.4
#links	546430	546415	659.94	659.92	611.38	611.33
#overflow	403843**4**	403843**0**	4877.**3**	4877.**2**	5074.**2**	5074.**0**
#tracks	5861**0**	5860**9**	70.78**5**	70.78**4**	48.51**3**	48.51**1**

On the way

- Many small problems with very recent OS releases (FC18, Ubuntu 12)
- Kernel updates a bit more complicated, every platform needs specific patches, because there is no "ARM-PC"
- Obviously problems with closed-source, x86-only, software (NeuroBayes) *
- Delicious architecture specific problems for the conaisseur
  - x86-icisms (e.g. sizeof empty struct)
  - Bad instructions issued by compiler (refused by assembler), toolchain problem?
- ROOT (v5.34.05) cintex not working completely (test fails), but subset required by LHCb seems to work.
- Loads of patches left, right and center due to newer compiler, boost, etc. . . versions

Building

- Compile-times are long on ARM. make -j 4 build of ROOT about 30 minutes on Viridis
- x-compiler chain has been setup.
- Problem with “fancy” builds creating and **using** intermediate binaries
- Can fix this with tweaking binfmt and using a VM
- In practice x-builds are painful for this stack, lots of room for improvement

Summary

- ARM is clearly a “hype” - we wanted to do a reality check
- Ported entire LHCb software stack to ARM (v7) on FC18
- Results are correct (up to small rounding errors)
- Single core performance about a factor 5 to 7 slower (compared to now obsolete Nehalem μ-architecture!)
- ARM memory bandwidth does not scale (yet)
- Power promise holds
- ARM-based microservers still need a strong TCO argument to be interesting

What's next?

- Improve multi-architecture build (cmake)
- Prepare maintainable, automatised cross-build system
- Wait for establishment of 64-bit ARM micro-servers, redo tests
- Compare with upcoming improved Intel micro-servers

Thanks

- Thanks to all members of the SFT group who supported us
- and to Boston HPC, UK, who kindly provided remote access to their Viridis platform